Performance Technical Report with SF-298

Feasibility Studies of

Nearest Neighbor Residual Vector Quantizer Classifiers for a

Collection of Signal and Sensor Waveforms:

# Automatic Target Recognition in SAR Images

Prepared by

Christopher F. Barnes and Byron M. Keel
Georgia Tech Research Institute
Georgia Institute of Technology
Atlanta, GA 30332–0857



Prepared for

Office of Naval Research
Ballston Centre Tower One
800 North Quincy Street
Arlington, VA 22217–5660



ONR Grant No: N00014–96–1–0308
GTRI Project No: A–5138–000

January 1, 1998

**GEORGIA INSTITUTE OF TECHNOLOGY**
**A Unit of the University System of Georgia**
**Atlanta, Georgia 30332**

Georgia Tech Research Institute

SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS<br>UNCLASSIFIED |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION / AVAILABILITY OF REPORT<br>APPROVED FOR PUBLIC RELEASE |
| 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S)<br>A-5138-000 | 5. MONITORING ORGANIZATION REPORT NUMBER(S)<br>A-5138-000 |
| 6a. NAME OF PERFORMING ORGANIZATION<br>Georgia Tech Research Institute<br>6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION<br>Office of Naval Research, Code 313 |
| 6c. ADDRESS (City, State, and ZIP Code)<br>Georgia Institute of Technology<br>Atlanta, GA 30332 | 7b. ADDRESS (City, State, and ZIP Code)<br>Ballston Center Tower One<br>800 N. Quincy Street<br>Arlington, VA 22217-5600 |
| 8a. NAME OF FUNDING / SPONSORING ORGANIZATION<br>Office of Naval Research<br>8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c. ADDRESS (City, State, and ZIP Code)<br>Code 313, Ballston Center Tower One<br>Arlington, VA 22217-5600 | 10. SOURCE OF FUNDING NUMBERS N00014-96-1-0308<br>PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO. |

11. TITLE (Include Security Classification) Feasibility Studies of Nearest Neighbor Residual Vector Quantizer Classifiers for a Collection of Signal and Sensor Waveforms

12. PERSONAL AUTHOR(S) Christopher F. Barnes and Byron M. Keel

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| Interim | FROM 1/1/97 TO 1/1/98 | 1 January 1998 | 71 + vi |

16. SUPPLEMENTARY NOTATION
Report covers second year of a three year grant.

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | ATR, SAR, detection, classification, MSTAR |
| | | | |
| | | | |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Reports results obtained from using direct sum successive approximations for detecting and classifying targets in SAR imagery.

| | |
|---|---|
| 20. DISTRIBUTION / AVAILABILITY OF ABSTRACT<br>☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL | 22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL |

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

# Acknowledgment

This report summarizes the current status of ongoing work in a Georgia Tech Research Institute (GTRI) research project entitled, *"Feasibility Studies of Nearest Neighbor Residual Vector Quantizer Classifiers for a Collection of Signal and Image Sensor Waveforms."* This research is supported by the Office of Naval Research (ONR) Grant N00014–96–1–0308, and is being performed under GTRI Project A–5138. This interim Performance Technical Report covers the second year of this three year grant, and reports on work performed between 1 January 1997 and 31 December 1997. The research focus area of this report is on the use of GTRI's classifier for automatic target recognition in synthetic aperture radar (SAR) imagery.

The ONR technical project director is James R. Buss. The GTRI investigators are Christopher F. Barnes, PhD, and Byron M. Keel, PhD. The investigators express appreciation to ONR for its support and to Mr. Buss for his oversight in the execution of this research program.

# Data Source Credits

The synthetic aperture radar data used in this research was obtained from Moving and Stationary Target (MSTAR) data sets. MSTAR measurements were sponsored by the Defense Advanced Research Projects Agency (DARPA). The data was provided via courtesy of Wright Laboratory's Integration Branch (WL/AACI). Information about MSTAR data is available at http://www.mbvlab.wpafb.af.mil/public/MBVDATA.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Executive Summary

This executive summary contains a concise overview of the grant purpose, problem statement and proposed solution, the research objective, and the technical approach used to achieve this objective. Experimental setups, performance results, and conclusions are also summarized.

## 1.1 Grant Purpose

The purpose of this ONR grant is to support the evaluation of the performance of a particular joint compression/classification algorithm called *nearest neighbor residual vector quantizer (NN-RVQ) classification* on data obtained from a variety of sensor types and for a variety of applications. NN-RVQ is based on a recent mathematical development called *direct sum successive approximations (DSSA)*. DSSA can be used as a technical foundation for data compression or pattern recognition algorithms, or for a single algorithm that does both. DSSA uses an unconventional mathematical data analysis/synthesis process to construct structured pattern dictionaries that can be efficiently searched (in terms of computation and memory). These patterns can be used as codevectors in vector quantizers (VQs) used for data compression, and as templates in nearest neighbor classifiers used for data classification. The purpose of this grant is to assess the performance of NN-RVQs when they are used for classification, compression, or joint classification and compression of various types of sensor data.

## 1.2 Problem Statement

There are two underlying problems addressed by this research:

1. **The Data Classification Problem:** The excessive computational resources required for real-time classification of data onboard sensor platforms. The need for real-time

classification is motivated by the requirements for an onboard data-prescreen capability for discriminating a general class of targets from clutter, or a complete onboard target recognition capability that is capable of identifying specific targets or threats. Solutions to the real-time, onboard, data classification problem seek to maximize classification performance when the algorithm is restricted in memory and computational complexity.

2. **The Data Compression Problem:** The lack of sufficient bandwidth required to transmit data at a high rate from a remote sensor to the data user, or equivalently, the lack of sufficient computer memory required to store large volumes of measured sensor data. Solutions to the data compression problem seek to represent data with more efficient binary representations.

## 1.3 Proposed Solution

GTRI proposes to use DSSA as the basis for algorithms that can be used for target recognition, for data compression, or both in memory and computation restricted signal processing algorithms that solve the data classification and compression problems. The DSSA design process is similar to the K-means algorithm often employed in clustering data and constructing exemplars for use in nearest neighbor classifiers, and for constructing codevectors for use in vector quantizer data compression algorithms. The primary difference between the K-means algorithm and the DSSA clustering algorithm is that DSSA provides a relatively small set of exemplar basis functions that can be used to form a much larger set of "structured" templates for nearest neighbor classification. The templates are the direct sums of variable numbers of over-determined basis functions that are formed by adding a new basis function one stage at a time in a memory- and computation-efficient manner. This progressive process of building templates forms a sequence of successive approximations of measured sensor data. GTRI's technical approach is to use DSSA basis functions to form templates for use as templates in nearest neighbor classifiers, and for use as codevectors in vector quantizers.

## 1.4 Research Objective

The object of this research is to determine the feasibility of performing DSSA-based compression and/or classification of data from a variety of image and signal sensors. This grant has been structured by ONR to permit flexibility as to exactly what sensors, data, and applications are evaluated by GTRI. The use of DSSA for classification does not require feature extraction[1]. DSSA may be incorporated into NN-RVQ classifiers in such a way that direct classification of data samples is possible. Thus, DSSA classifiers can be easily and

[1]This does not preclude the use of a DSSA system as a conventional discriminate that operates on a set of extracted data features.

automatically designed for a wide variety of sensor types—all that is required is sample data for training and evaluation purposes. The ease at which DSSA classifiers can be designed and implemented allows a wide variety of sensors and associated data to be investigated in a cost efficient manner. Candidate sensors include defense related imagery and signal sensors, and sensors that support dual-use applications.

GTRI investigated in 1996 the performance of DSSA for a dual use application: computer assisted diagnosis and compression of medical mammography image data [1].

GTRI investigated in 1997 the performance of DSSA for a defense application: air-to-ground target detection and identification using synthetic aperture radar (SAR) data. This Interim Technical Reports describes the results of this phase of the research grant.

GTRI will be responsive to any directive from ONR as to which data sets should be tested and evaluated in this research project in 1998 (to the extent that funding levels and data availability permit). Possible data sets for 1998's effort include electro-optical (EO) data, infrared (IR) data and multispectral (MS) data. Classification tasks associated with EO/IR/MS images include target detection and land-use classification. GTRI is also willing to extend this year's SAR investigation to address the detection and identification of subtarget features for targets partially obscured in revetments or foliage.

## 1.5 Experiment Overview

The performance of the DSSA classifier was evaluated on SAR imagery containing targets and clutter at three resolution levels: $4 \times 4$ foot resolution cells (herein referred to as "low" resolution data), $2 \times 2$ foot resolution cells (herein referred to as "medium" resolution data), and $1 \times 1$ foot resolution cells (herein referred to as "high" resolution data). The SAR data set contains data measured from three targets—a tank, an infantry fighting vehicle, and an armored personnel carrier. DSSA-based detection and classification were performed on the SAR data in a three stage approach where each stage used data at a different resolution level. DSSA-detection processing was first performed on the low resolution data. DSSA-classification processing was then performed on all detected objects at the medium resolution data. The regions-of-interests (ROIs) at the medium level that were not classified with sufficient confidence were processed again with a DSSA-classifier at the high resolution level to reach a final classification decision. All experiments were conducted using essentially raw SAR pixel data as feature data (simple smoothing was performed on the data before detection and classification).

Although this progressive execution of DSSA-detection followed by DSSA-classification may make it appear as though the two algorithms are different—they are not. DSSA-detection seeks to discriminate between clutter and a general class of targets, i.e., detect targets. DSSA-classification seeks to discriminate between separate target classes, i.e., classify targets. The algorithm used in either case is the same—only the end objective differs.

| Resolution Level | Probability of Detection | Probability of Classification | False Alarms per Image |
|---|---|---|---|
| Low | 99.70% | 69.30% | 5.07 / Image |
| Medium | Not Applicable | 92.74% | 0.16 / Image |
| High | Not Applicable | 98.75% | 0.00 / Image |

Table 1.1: Summary of detection, classification, and false alarm rates.

## 1.6 Performance Summary

Table 1.1 presents a summary of the detection, classification,, and false alarm results for the three target-class problem. A total of 1365 targets were contained in the test set, 1361 were detected at the low SAR resolution level (99.70%), and of these 1361, a total of 1344 were correctly classified at the medium and/or high SAR resolution levels (98.75%). One hundred SAR images containing rural and suburban clutter scenes were also tested to estimate the false alarm rate. Each image covered about one-tenth of a square kilometer. An average of five false alarms per image occurred at the low resolution level, an average of one false alarm in six images occurred at the medium resolution, and no false alarms were detected in the 100 test images at the highest resolution level.

## 1.7 Conclusion Highlights

These experimental results gave nearly perfect detection and classification results. However, these results should be viewed as overly optimistic for the following reasons. First, a test case with only three target classes was conducted with no "confuser" classes included in the experiment. Confusers are target-like vehicles that are in fact not military targets. The DoD has restricted data for a total of 20 targets and 5 confusers, GTRI is trying to acquire copies of this larger test set for future work. Second, a suite of simple constant false alarm rate (CFAR) algorithms were used to minimize the false alarm rate. The parameters used in the CFAR algorithms were optimized for the test target data, but no test clutter data were used in selecting the CFAR parameters. Nevertheless, these experimental results demonstrate that DSSA detection and classification of SAR data is promising and deserves additional investigation.

This report establishes that it possible to detect and classify SAR data with the use of SAR pixel data as features. Thus, training-on-the-fly and new-target-extension of DSSA based algorithms would be easy processes not requiring extensive non-recurring engineering cost for new target feature definition and extraction. This report further demonstrates that DSSA may hold computational advantages for the following reasons. 1) High resolution SAR image formation is not required for all measured SAR phase history data. Only the detected ROIs are processed with advanced SAR image formation algorithms to achieve higher resolution. 2) DSSA enables the formation of a combined target detection and classification algorithm

architecture that is homogeneous; the DSSA-classification algorithm is basically the same as the DSSA-detection algorithm. 3) DSSA is easily parallelized and pipelined for efficient real-time implementations.

# Chapter 2

# Introduction

This introductory chapter provides background information and gives an overview of the approach used by GTRI to construct algorithms based on DSSA for detecting and classifying target signatures in SAR images.

## 2.1 Automatic Target Recognition

This section provides an overview of the military objectives, tactical strategy, and technical approaches and goals set forth by the U.S. Department of Defense (DoD) for the purpose of developing effective automatic target recognition.

### 2.1.1 Military Objectives

The Department of Defense is investing automatic (or assisted) target recognition (ATR) systems to support the future military requirements of the joint war fighter's operational needs. The war fighter is seeking to obtain dominant battlefield awareness with real-time identification of targets. Combat identification beyond visual range aids survivability, effective weapon employment, and reduced fracticide. The detection and identification of time critical targets such as ground based missile launchers is vital in the theater front and in the littoral fighting environment of the U.S. Navy. In urban terrains, the detection and recognition of high valued targets in high clutter backgrounds is needed to support precision guided weapons to reduce collateral damage. A complete battlefield awareness and data dissemination system combines information from signal intelligence, terrain maps, national intelligence assets, national information repositories, global weather information, EO/IR/MS sensors, and radar (moving target indication (MTI) and SAR) positioned tactical reconnaissance aircraft and UAVs and numerous other sources.

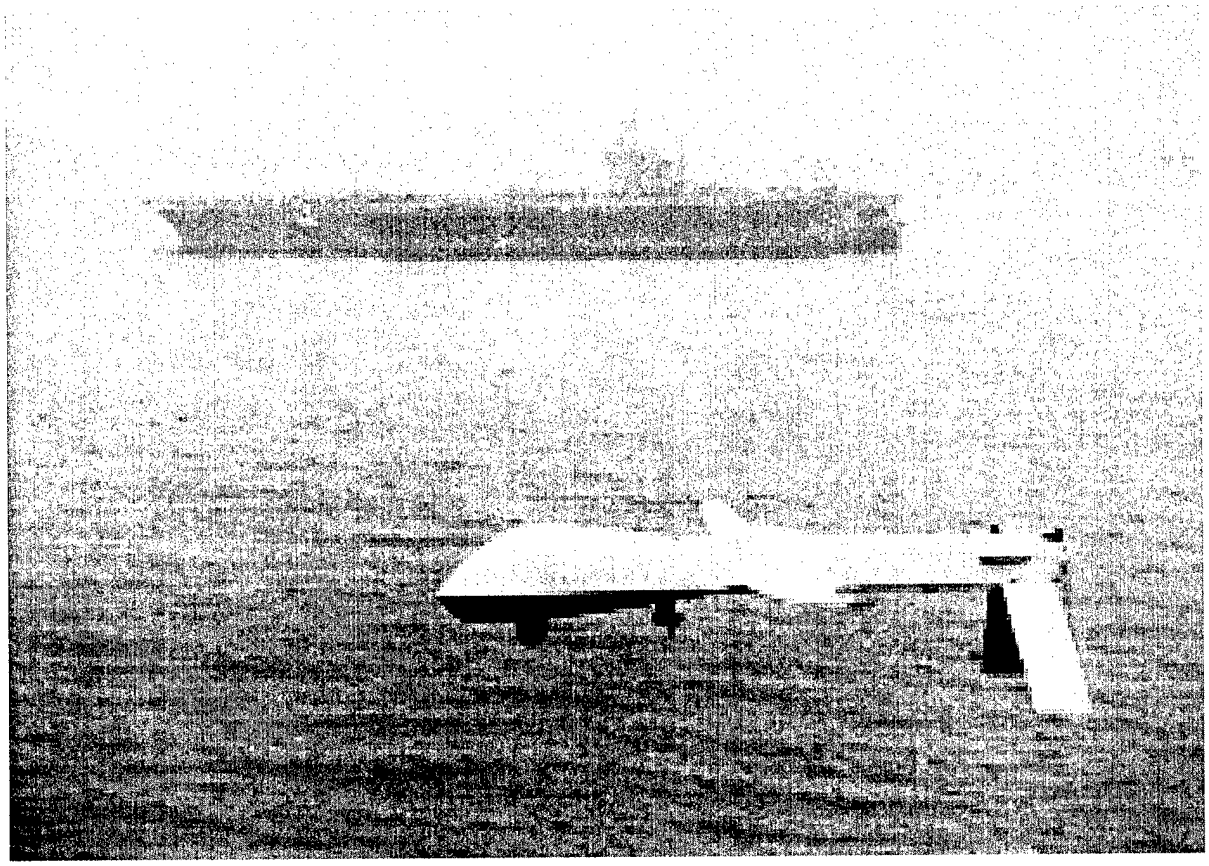

Figure 2.1: Predator UAV.

### 2.1.2 Battlefield Awareness

With the advent of the information age and the development of high resolution sensors, the military has placed much attention and resources in the area of comprehensive and timely battlefield awareness. The goal is to detect, identify, and track all vehicles in order to define the ground order of battle and missile order of battle targets. The sensors used to extract available information include electro-optical sensors, infrared sensors, and synthetic aperture radars. The sensor assets are positioned on surveillance aircraft such as the U-2 or on emerging platforms such as high-altitude endurance unmanned aerial vehicles (HAE UAVs) (e.g., current UAVs include Predator (Figure 2.1) and Hunter (managed by the Navy's Program Executive Officer for Cruise Missiles and Joint Unmanned Aerial Vehicles). Future UAVs include Global Hawk, Dark Star, and Outrider. The UAVs are used to reduce the risk to the war fighter and to increase the time over which data can be gathered. An example of the tactical UAV environment is illustrated in Figure 2.2.

At the conference on 21st Century Investment Strategy for Airborne Reconnaissance Sensors, headed by General Kenneth Israel, the war fighter requirement for imagery data was stated to be on the order of 40,000 square nautical miles / day at a one foot resolution. This is a monumental task which is likely to require detection processing at lower resolutions with

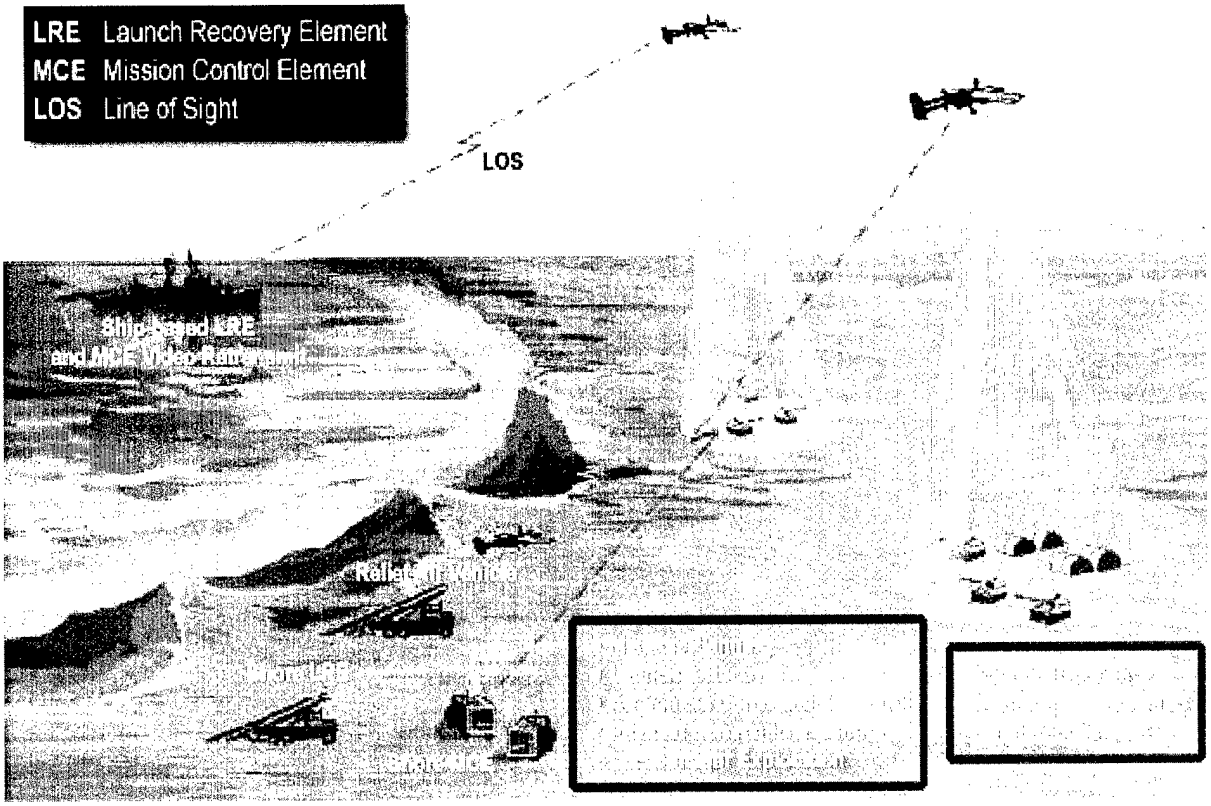


Figure 2.2: UAV deployment in littoral environment.

classification processing at higher resolutions images or subregions of interest.

### 2.1.3 Surveillance Goals

Modern surveillance systems are required to monitor large volumes of space while detecting, identifying, and tracking all militarily significant targets. In order to achieve these requirements, the first stage in any imagery dominated ATR algorithm focuses on identifying regions-of-interest using either low resolution imagery to reduce the computational load, or algorithms designed to efficiently partition the imagery into clutter regions (e.g., trees, grass, roads, water, etc.) and potential target regions. The probability of detection must be high at this stage, however, the false alarm rate may also be high at this stage since false alarms can be eliminated in latter stages through subsequent processing. The second stage then focuses on identifying targets within the regions-of-interest. High resolution data is applied at this stage to reduce the false alarm rate and to maximize the probability of identification. The identification stage of the ATR algorithm compares the measured data to either templates derived from measured data or synthetic data derived from target models, or extracts features from the measured data that are then compared with features associated with measured or synthetic template data. Both approaches require time to train the ATR

algorithms. The DoD has specified performance level goals for modern ATR systems: a probability of detection goal of 0.90 and probability of identification goal of 0.7 (specified in the DARPA funded MSTAR program). The false alarm goal is one per 1000 $km^2$.

## 2.2 GTRI's Technical Approach

GTRI's technical approach is the use of a novel nearest neighbor classifier with templates constructed with a new type of basis functions called *direct sum successive approximation* (DSSA) functions. This section provides an overview of the history and approach used by GTRI to develop novel DSSA classification systems. Chapter 3 describes the algorithm in detail and compares its structure with that of neural net classifiers.

### 2.2.1 Development History of DSSA Classifiers

The unoptimized structural architecture of DSSA originated in the technical area of data compression of images, and was initially called *multiple stage vector quantization* (MSVQ) [2]. This architecture was later optimized in two independent PhD thesis works [3, 4]. The optimized version of MSVQ, when applied to image data compression, is sometimes called *residual vector quantization* (RVQ) [5, 6, 7, 8].

The process developed by one of the investigators for optimizing the DSSA structure was awarded a patent in 1993 and assigned to Brigham Young University (BYU) [9] (this patent was filed 14 August 1989), other related intellectual property is held by the Georgia Institute of Technology (GIT).

The technical areas of vector quantization (VQ) data compression and nearest neighbor (NN) data classification are closely related. Indeed, the encoder of a vector quantizer is identical to the conventional $k$-means NN classifier [10, 11, 12, 13]. The $k$-means clustering algorithm [14] used for generating exemplars for NN classifiers is also identical to the generalized Lloyd algorithm (GLA) [15, 16], also called the Linde, Buzo, Gray (LBG) algorithm [17], used for generating the codevectors of VQ codebooks [18]. Many researchers are currently exploiting the synergism between developments in VQ compression and NN classification (for examples, see [19, 20, 21]).

Researchers at GTRI have explored the application of RVQ to the nearest neighbor classification problem since 1989. The underlying theoretical concept of RVQ, whether applied to data compression or classification is the *direct sum successive approximation basis function* (DBF). This function is a multivariate function that can be used to represent samples of either one-dimensional time series (e.g., acoustic) data, or two-dimensional spatial (e.g., image) data.

GTRI investigated the feasibility of using DSSA to classify mines in sonar images under a previous ONR program (N61331-93-K-0035), performed between July 1993 and November

1994 [22], and investigated DSSA classification performance on acoustic backscatter for long-range mine detection (N61331-96-C-027) [23]. GTRI has also contracted with a major defense organization to continue investigations of DSSA applied to both side looking sonar (SLS) imagery, and forward looking sonar (FLS) data.

### 2.2.2 Novel Aspects of DSSA Classifiers

The most novel aspect of DSSA classifiers is their computation and memory simplicity relative to the number of exemplars searched for nearest neighbor classification. This relative simplicity permits nearest neighbor classification of feature vectors with many embedded features or "dimensions". For example, a classifier that is designed to classify only on the basis of a target's width and height, would likely have its performance improved if additional features such as texture, brightness, shadow size, etc., are added to the classification process. But most classification architectures become impractical when the feature vectors become large. In addition, two persistent problems have plagued most classifiers when high dimensional feature vectors are used. Collectively, these problems are referred to as the "curse of dimensionality":

**Design Phase Problem:** The problem of generating a large numbers of high dimensional exemplars with limited training data during the design phase of the classifier.

**Run Phase Problem:** The problem of classifier robustness when data not well represented by training data is encountered during the run mode of the classifier.

#### The Curse of Dimensionality in the Design Phase

The first problem is directly related to the number of parameters, or "degrees-of-freedom" (DoF) that must be specified when building a classifier. Typically, the number of degrees-of-freedom expands exponentially as the dimensionality of the classifier increases. Nearly all parametric classifiers have been so structured to combat the curse of dimensionality *by restricting the number of dimensions and by imposing structural constraints on the boundaries between decision regions.* Examples include linear discriminates, quadratic discriminates, and neural nets (NNets), which generally manipulate parametric hyperplane, quadratic, or ellipsoidal decision boundaries. The problem with constraining the form of decision boundaries in high dimensional spaces is that the number of "facets" associated with decision regions often increases exponentially as dimensionality is increased.

Unconstrained nearest neighbor classifiers also suffer from the design phase problem when the number of exemplars that must be generated becomes large. A large number of exemplars requires a large training set, and since, as a rule-of-thumb, the number of required exemplars increases with increasing feature vector dimensionality, practical nearest neighbor classifiers have restricted dimensionality.

DSSA classifiers are not subject to the design phase part of the curse of dimensionality. Unlike parametric classifiers, DSSA classifiers seek not to directly constrain the dimensionality nor decision region boundaries to avoid the curse of dimensionality, *but structurally constrain the content of the exemplars by limiting the fidelity of the digital representations of the feature amplitude values.* The exemplar amplitude values are restricted to be only those that can be constructed by a sequence of DSSA basis functions. The number of permitted DSSA basis functions available at each stage of the exemplar synthesis process can be as small as two, and is often no larger than twenty. The use of multiple stages of templates with small numbers of templates at each stage has a tremendous impact on reducing the amount of training data that is required to generate templates for high dimensional feature vectors. The DSSA design process requires that only one stage of the DSSA basis functions be generated (or improved) at a time—thus, the entire training set need only be partitioned between the small number of DSSA basis functions that exist at a single stage. The DSSA design process is practically never starved for training data at the stage level.

Although the number of DSSA basis functions that exist at each stage is usually quite small, the number of direct sum exemplars that can be formed by the DSSA basis functions increases exponentially with increasing number of DSSA stages. For example, if the number of DSSA basis functions at each stage is N, then the number of direct sum exemplars $\mathcal{N}$ available to the nearest neighbor classifier is $\mathcal{N} = N^P$ for a $P$-stage system. The key to the DSSA approach is that only a small subset of these possible direct sum exemplars are constructed during the search process for each input vector, and this construction takes place in real-time as dictated by the contents of each input feature vector. Thus, an exhaustive search over an enormous, static, prestored database is not required.

### The Curse of Dimensionality in the Run Phase

Of course, there are limits to how much information can be gleaned for a classification process from limited training data. If training data is severely limited, not all of the direct sum exemplars will be effective in the classification process if large numbers of DSSA stages are generated. Two questions remain: (1) how many DSSA stages should be designed for a given training set size, and (2) at what minimum training set size for a given feature dimensionality is DSSA performance acceptable? The answer to the first question is known and is explained in Section 3, the general answer to the second question is currently unknown and must be addressed empirically in each specific case.

### Is the Curse of Dimensionality Omnipresent?

Traditionally, in more conventional nearest neighbor classification systems, rather demanding rules-of-thumbs have required minimum training set sizes on the order of 10-100 training vectors for every template [24, 25, 26]. However, recent research by others has started to question the transcendent nature of the "curse of dimensionality", and some experimental results are appearing in the literature that suggest that high dimensional nearest neighbor

classifiers can still be designed in certain cases with limited training data and yet obtain good performance [27, 28]. The results of this research also suggest this is possible (see Chapters 4 and 5).

### The Use of Raw Features

A primary focus of this research is the use of raw SAR pixel data in a target classifier—higher order feature definition and extraction is a secondary issue. The feature set used in this research is SAR image pixel data that has only been slightly preprocessed to reduce the level of speckle. The feature vectors are large—they are formed from the pixels of two-dimensional "snippets" extracted from the SAR imagery. The snippets used in this research are as large as $41 \times 65$ pixels and contain as many as 2,665 pixels. These large-but-simple feature vectors essentially contain the entire SAR radar signature of each target.

### The Use of Limited Training Data

The use of raw SAR data has one important advantage: DSSA systems can be easily extended and updated to accommodate new threats. *The feature definition, extraction, and classifier design process can be entirely automated, and extension of the classifier to a new target class does not require human intervention.* Thus, training-on-the-fly and rapid updating and insertion of new targets into a DSSA classifier is a straight forward process.

Of course, the more training data that is available—the better. A DSSA system that is trained-on-the-fly with limited training data will not perform with the same level of confidence as a system with exhaustive amounts of training. However, DSSA classifiers provide, in essence, an exhaustive search of all available training data during the on-line classification process. Thus, a DSSA system with limited a priori target signature exposure will provide classification results consistent with all available data. Moreover, DSSA classification decision are not necessarily binary—DSSA conveys confidence about its decisions. Thus, in assisted target recognition applications where training-on-the-fly is most appropriate to counter new threats that are not understood very well, the level of confidence (or lack of) of the DSSA detections/classifications is also conveyed to the human operator.

The results of this research and other similar work done by GTRI show that effective DSSA classifiers can be designed with limited training data, and that DSSA is able to deal with large feature vectors and large intraclass signature variability. Thus, the DSSA approach permits the development of a SAR target detection and classification system that is flexible (can be updated to accommodate new threats), is robust (works with limited training data), and provides good performance.

## 2.3 Related Classification and Compression Literature

A VQ data compression system partitions a data space and assigns each cell to a codevector. A classifier with nearest neighbor templates partitions a decision space and assigns the region about each exemplar to a class. A difference between the two problems is related to the design procedures; a VQ is most often designed to minimize compression distortion, while a classifier is designed to minimize classification error.

A literature search was conducted to determine what approaches have been used by other researchers to design such compression-classifiers. Four related research areas were found in the literature. One is the use of vector quantization for pattern recognition [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 36, 39, 40, 41]. The second is compression of data that are subsequently processed for target detection [42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53]. The third is sequential detection theory [54, 55, 56], and the fourth is encoding of multiple correlated observations for joint detection processing [57, 58, 59].

## 2.4 Report Organization

This scientific and technical report is organized as follows.

**Chapter 1** (Executive Summary) explains the problem addressed by this research project and provides a summary of the research objectives, technical approach, and experimental results. Concise summaries of GTRI's conclusions and recommendations are also given.

**Chapter 2** (this introductory chapter) contains an introduction that explains the target detection problem associated with SAR target detection and classification, provides background material on GTRI's technical approach.

**Chapter 3** gives a detailed description of the DSSA classifier. Related material on neural nets is included.

**Chapter 4** describes the SAR database, the data preprocessing steps, the classification goals, and experimental results. This chapter also describes the computational and memory costs required to implement the DSSA classifier.

**Chapter 5** gives GTRI's conclusions and recommendations related to DSSA classifier performance and implementation, and also contains a set of suggested topics for further research and development.

# Chapter 3

# Automatic Target Recognition

GTRI has examined and compared the structures of DSSA and various types of neural networks and has found similarities between DSSA classifiers and the structure of radial basis function (RBF) neural networks. Although DSSA originated in the design and application of vector quantization data compression, this report describes DSSA in terms of the structural components of neural networks to make this work more easily accessible to a wider audience than just the data compression community. A previous description of DSSA in terms of vector quantization is in [22]. It is important to point out that the design of DSSA classifiers was not motivated in any way by the development of neural networks, but will be simply described in this report in terms also used to describe the architecture of neural networks.

## 3.1 RBF Neural Network Classifier

A radial basis function neural network consists of a set of sensory units or source nodes that form the input layer, a hidden layer of computational nodes, and an output layer of computational nodes. An architectural diagram of a RBF network is shown in Figure 3.1. The adjacent layers are exhaustively interconnected and the input signal propagates through the network in a forward direction. The network has $k$ nodes in the input layer, one for each of the elements of the input feature vector, $N$ nodes in the hidden layer, and $M$ nodes in the output layer, one for each of the possible classification decisions. The first-layer connections are not weighted, thus each hidden layer node receives an unaltered copy of the input feature vector.

Associated with each node of the hidden layer is an exemplar $\boldsymbol{y}_h$ that is a centroid of some portion of the neural net training data. The hidden nodes collectively represent all of the training data with centroid-based approximations which are called radial basis functions, or RBF-centroids. Each hidden node takes the input feature vector and computes the distance from its RBF-centroid and applies a nonmonotonic transfer function to produce a continuous, positive output activation level. The second layer's connections are weighted and summed in the output nodes. The activity levels of the nodes in the output layer are interpreted as

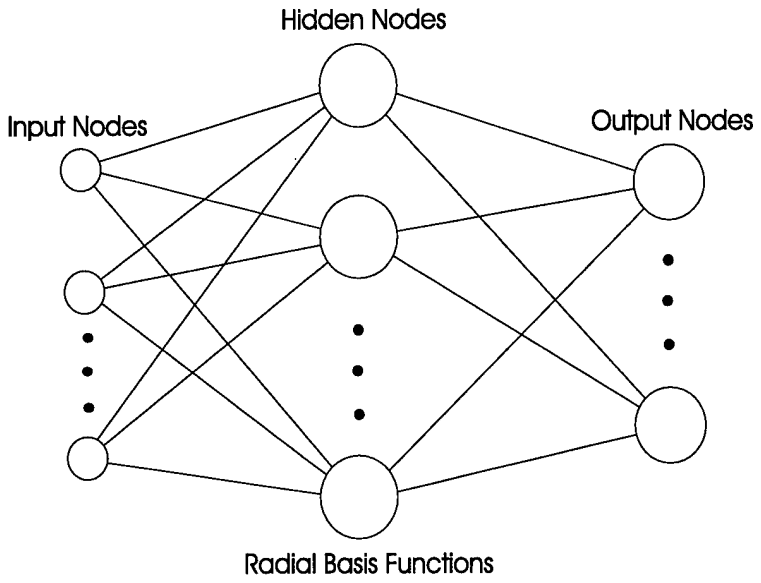


Figure 3.1: A class-independent, RBF-centroid neural network architecture.

the unnormalized likelihoods of the corresponding classes, and the class of the output node with the highest activation level is taken as the classification decision.

The transfer function of the $N$ hidden nodes is similar to the Gaussian density function given by

$$a_n = \exp\left[-\|\boldsymbol{x} - \boldsymbol{y}_h\|^2/\sigma^2\right] \tag{3.1}$$

where $a_n$ is the activation of the $n$th RBF in the hidden layer given the input feature vector $\boldsymbol{x}$. The distance scaling parameter $\sigma$ determines over what distance in the feature space a RBF-centroid will have significant influence. The output nodes of the neural net compute the $m$th class activation level by

$$\boldsymbol{z}_m = \sum_{n=1}^{H} w_{mh} a_h + \theta_m \tag{3.2}$$

where $\theta_m$ is a class bias constant, and the $w_{mh}$ are the weights applied to the outputs of the hidden layer.

The training elements that comprise the training data for a RBF neural net consists of associated pairs $\{(\boldsymbol{x}_l, \boldsymbol{c}_l);\ l = 1, 2, \ldots, L$ of feature vectors $\boldsymbol{x}_l$ and associated class label vectors $\boldsymbol{c}_l$, that are length $M$ vectors with a value of 1 in the position corresponding to the correct classification of $\boldsymbol{x}_l$ and zeros elsewhere. The network is trained by adapting the weights to minimize

$$E(\boldsymbol{y}_h, \sigma, w_{mh}) = \sum_{l=1}^{L} \|\boldsymbol{z}_l - \boldsymbol{c}_l\|^2 \tag{3.3}$$

the sum-squared-errors between the network outputs $\boldsymbol{z}_l$ and the target values $\boldsymbol{c}_l$ over the set of training examples $\boldsymbol{x}_l$.

### 3.1.1 RBF Neural Net Design Method

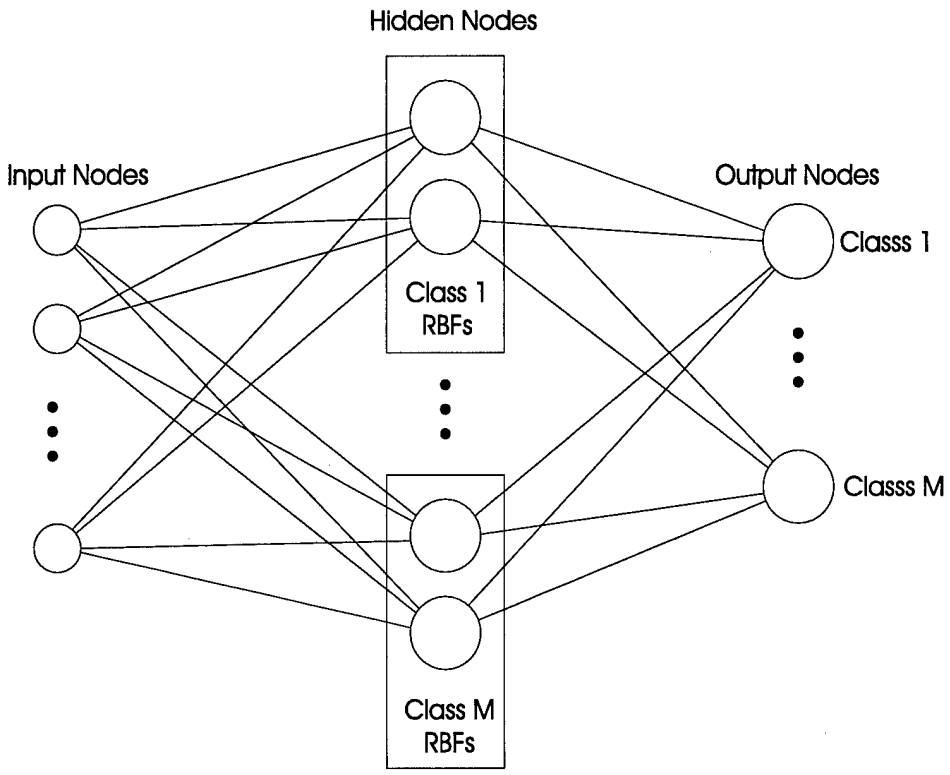


Figure 3.2: A class-dependent, RBF-centroid neural network architecture.

Various methods have been proposed for training neural network classifiers. Burton and Lai [60] trained their neural network using four steps:

1. A fixed number $N$ of RBF-centroids was selected for the hidden layer.
2. The centroids $\boldsymbol{y}_h$ were determined by the same design technique used for constructing codevectors/exemplars used by learning vector quantizers (LVQ) [19, 21].
3. The scaling parameter $\sigma$ was determined by a nearest neighbor heuristic.
4. The weights of the second layer of connections were determined by minimizing the mean-squared-error between the computed and desired output of each output node.

In actuality, Burton and Lai used a separate scaling parameter $\sigma_k$ for each element of the feature vectors. Thus, the radial basis functions became ellipsoidal basis functions (EBF) in their implementation. Furthermore, they made the EBFs class-dependent, that is, each EBF centroid was formed by using training data from only a single class. They note that this simplifies the network, provides for easier training, and allows for easier addition of new classes without affecting existing class basis functions. The structure of the class-dependent EBF network is shown in Figure 3.2.

### 3.1.2 LVQ in RBF Neural Net Classifiers

Kohonen [19, 61, 20] proposed a *likelihood or learning vector quantizer* (LVQ) to perform classification using a VQ encoder and codebook, where the encoder operates as an ordinary minimum mean squared error selection of a representative from the codebook, but the codebook is designed in a manner that attempts to reduce classification error implicitly rather than reducing mean squared error. Kohonen's algorithm is similar to Stone's [62], who constructed a general formulation of nearest neighbor methods for parametric regression, in which a general weighting dependent on class membership of several nearest neighbors were applied to the classifier.

Kohonen used a heuristics to argue that moving centroids according to nearby class membership should asymptotically have the effect of approximating a Bayes risk. His general goal was to imitate a Bayes classifier with less complexity than other approaches such as neural networks. Kohonen argued that for the case of Gaussian data, the partition induced by a VQ can approximate that required for a Bayes estimator—but this is a heuristic algorithm based on intuition [33].

Kohonen's approach has been widely used for classification of such disparate applications as the classification of speech sounds [35], of objects in clutter in synthetic aperture radar [38, 63], of proteins [36], of bird songs [39], of oceanic signals [37], and other applications [64, 32, 35].

## 3.2 DSSA Classifier

A structural diagram of a DSSA classifier is shown in Figure 3.3. There are four major points of difference between the RBF neural net and the DSSA classifier:

1. The RBF are not precomputed and stored, but are dynamically created on-the-fly as direct sum exemplars with the use of a prestored set of DSSA basis functions (DBFs).
2. The number of basis functions used for each input vector is not predetermined, but is data-dependent and varies between classes, and can even vary within a class for different feature vectors.
3. RVQ design methods are used to generate the DBF instead of the LVQ design methods used to generate the RBFs.
4. Output weights are computed differently and heuristic logic is (currently) used in the output layer.

The most obvious difference in comparing the architectures of Figures 3.2 and 3.3 is in the internal mesh of stages that comprise the hidden layer of the DSSA neural net. The hidden layer is a feed forward, fully connected mesh of DSSA basis function.

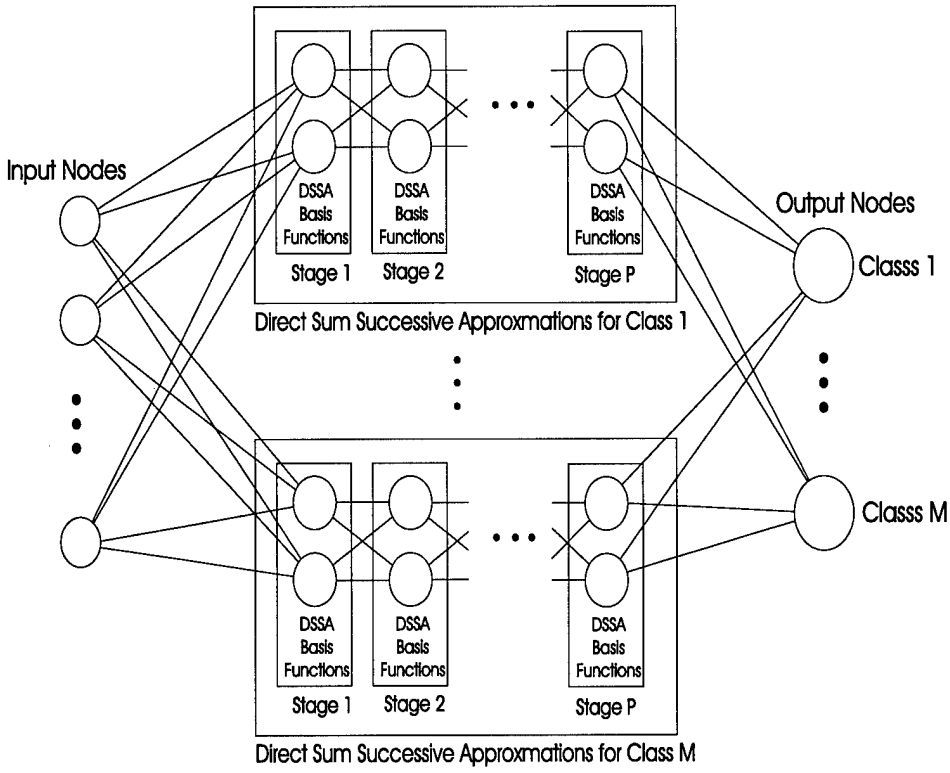


Figure 3.3: A class-dependent, DBF-centroid neural network architecture.

The DBF stages form a linear set of basis functions that construct direct sum exemplars. The early DBF stages in the sum represent the large amplitude, coarse features of the input vectors, while the later DBFs contain increasingly finer amplitude detail. The direct sum exemplars are formed and examined in a progressive manner until a direct sum exemplar is discovered that matches the input data with a predetermined fidelity threshold.

This matching process is performed within each of the class-dependent hidden layer systems. The system that provides the best match is declared as the classification decision.

The input vectors are not forced to propagate from the first DBF stage to the last, but may exit the system when confident classification decisions are made. By allowing a variable number of stages in the recognition process, the system can devote fewer computational resources to the "easy" problems, and more to the "hard" problems. For instance, this system first attempts to classify data represented by a coarse approximation. If the classification does not succeed with a high level of confidence, additional details are then added to the data representation such that a more accurate representation is obtained. Then the decision system tries once again to reach a classification decision with an acceptable level of confidence. This process is repeated until either it is determined that the DBFs do not span the space of the current input data (i.e., do not belong to the DBF class) or the data are "confidently" classified. This approach permits the matches to be less than stellar when the training data is limited, and thus helps the system to be robust. The matches may not be perfect, but from the given alternatives generated by the other classes, the correct class is

often the best match.

### 3.2.1 RVQ in DBF Neural Net Classifiers

Each DSSA-mesh interconnect path has dimensionality equal to the feature dimension. The weights of these paths are given by

$$\boldsymbol{x}_p = \boldsymbol{x} - \sum_{p=0}^{P} \boldsymbol{y}_p(\mathcal{E}(\boldsymbol{x}_p)) \tag{3.4}$$

where $\boldsymbol{y}_p(\mathcal{E}(\boldsymbol{x}_p))$ is the EBF result of a local nearest neighbor search operator $\mathcal{E}(\cdot)$ for $\boldsymbol{x}_p$ over the $p$th stage DBF set $\{\boldsymbol{y}_p(j);\ \ j = 1, 2, \ldots, N_p\}$. That is, each stage is searched in turn to find the most similar DBF to the $p$th stage input vector $\boldsymbol{x}_p$. The measure of similarity, or distance, used is the sum-of-squared-differences between the feature vector (for the first stage), or the causal residual vector (for all other stages) and the DBFs.

At the output of the first stage, the difference between the feature vector and the nearest DBF-centroid is formed to generate the first stage residual vector. This residual vector is then input into the second stage and then second stage nearest neighbor searches are conducted to find the best second stage DBF. This process is repeated for an arbitrary number of stages. If the resulting distance is large, the pattern match is poor, and if the distance is small, the pattern match is good. (It may be more appropriate to call this measure a dissimilarity measure.)

An approach similar to that given in [65] is used to determine early in the search process if the input data does not belong to the DBF class.

A RVQ design process is used to generate the DSSA stages. These DSSA stages usually only have a few, possibly just two, DBFs; this approach greatly expands the number of available direct sum exemplars that can be efficient searched. The number of possible direct sum exemplars that can be constructed from DBFs is

$$\mathcal{N} = N_1 \times N_2 \times \cdots \times N_P \tag{3.5}$$

where $P$ is the number of stages, and $N_p$ is the number of DBFs at the $p$th stage.

A normalized signal-to-noise fidelity criterion is specified before the RVQ design process begins, and this process generates the required number of DSSA stages necessary to meet specified fidelity criterion. The SNR fidelity used in the design process is defined as

$$w = 10\log_{10}\frac{\|\boldsymbol{x}_0\|^2}{\|\boldsymbol{x}_P\|^2} \tag{3.6}$$

$$= 10\log_{10}\left[\frac{\|\boldsymbol{x}_0\|^2}{\|\boldsymbol{x}_0 - \sum_{p=0}^{P}\boldsymbol{y}_p(\mathcal{E}(\boldsymbol{x}_p))\|^2}\right], \tag{3.7}$$

where $\boldsymbol{x}_0$ is the original feature vector input into the first stage.

### 3.2.2 DBF Neural Net Design Method

The training of the DSSA classifier consists of two parts. First the training data is processed by extracting feature vectors containing example SAR data from each target class. The RVQ training algorithm generates a set of DBFs at each stage of the DBF neural net that is able to progressively approximate the input feature vectors generated by a target class. If all these training vectors are sufficiently distinct, each one will produce a unique path through the DBF stages. These paths form a decision tree. For example, with 16 DBFs per stage, and 16 stages, $16^{16}$ distinct patterns could be generated, each with an unique path through the decision tree. However, to the extent that training data can be clustered, the number of distinct paths generated in the RVQ design process will be reduced. These paths are recorded in the form of a linked list of decision thresholds.

The thresholds associated with each direct sum exemplar are used to label decision regions within the decision space, these labels designate whether a given region of the decision space was explored during the training phase.

Full path thresholds provide the best performance and were tested in this report, but in some cases, these may have high implementation costs; partial path thresholds provide a tradeoff between performance and the memory required to store the decision tree. The threshold selected in this report is the maximum-in-class distance encountered during the training phase for each direct sum exemplar. This distance threshold determines the most dissimilar training target data that is used to construct the associated DBF. This threshold can be used to label decision regions within the decision space "target-like" or "unknown".

If the DBF classifier is tested on the training data, the threshold-based DSSA classifier will always provide perfect performance (100% PdPc - 0% FA) on all training data if the SNR fidelity level is selected to be sufficiently high.

# Chapter 4

# Experimental Results

This chapter provides a description of the SAR data set used by GTRI in the DSSA classification experiments, and the preprocessing applied to the data set before training and testing the DSSA-based ATR algorithm. Experiments that test DSSA detector/classifier performance for automatic target recognition in SAR imagery are described in detail. The implementation cost of DSSA and SAR image formation cost savings are also presented.

## 4.1 SAR Image Database

The Defense Advanced Research Project Agency (DARPA), in conjunction with Wright Labs, has developed the Moving and Stationary Target Acquisition and Recognition (MSTAR) program for the development of an ATR system capable of detecting and identifying time critical targets from two-dimensional SAR imagery. The program has directed the collection of SAR imagery on both US and foreign vehicles under a variety of conditions (e.g, varied configurations, articulations, obscurations, camouflage, and revetments). The program has to date collected SAR images on twenty targets and 700 clutter scenes consisting of rural and urban clutter. In an effort to maximize the research effort in this area, the MSTAR's program has released to the public a CD containing images from three foreign targets and 100 clutter scenes (both rural and urban). The target images were partitioned by Wright Labs into training and testing data sets for algorithm development and performance evaluation. The analysis in this report is based on applying the DSSA-based ATR algorithm to this set of data. The following sections will describe the data set in more detail.

### 4.1.1 MSTAR Target Images

The MSTAR public data set contains three targets: a T72 tank, a BMP2 infantry fighting vehicle, and a BTR70 armored personnel carrier. Three different T72 and BMP2 target configurations were provided in the data sets. The different configurations allow an ATR

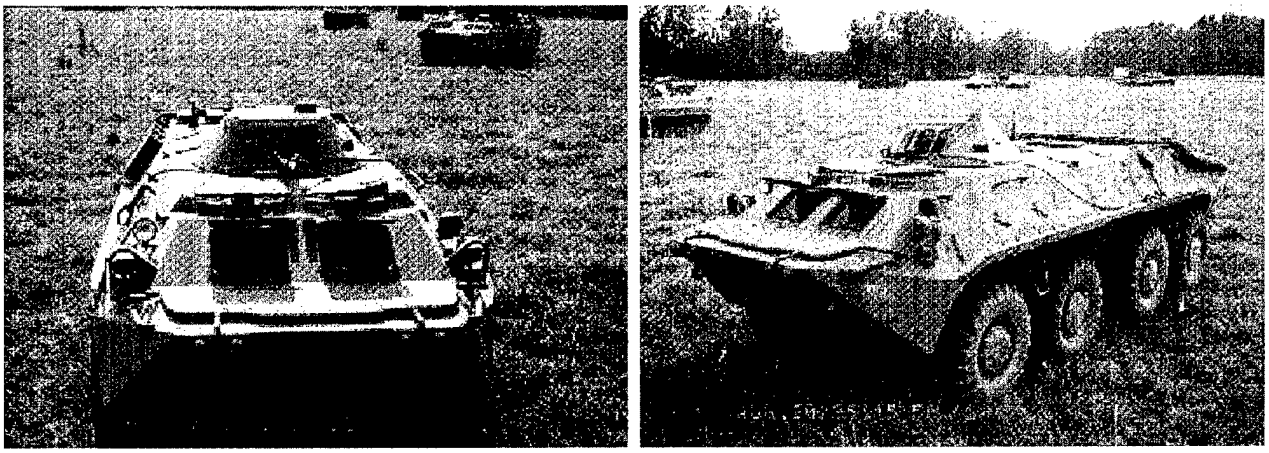

Figure 4.1: BTR70 armored personnel carrier in a C71 configuration.

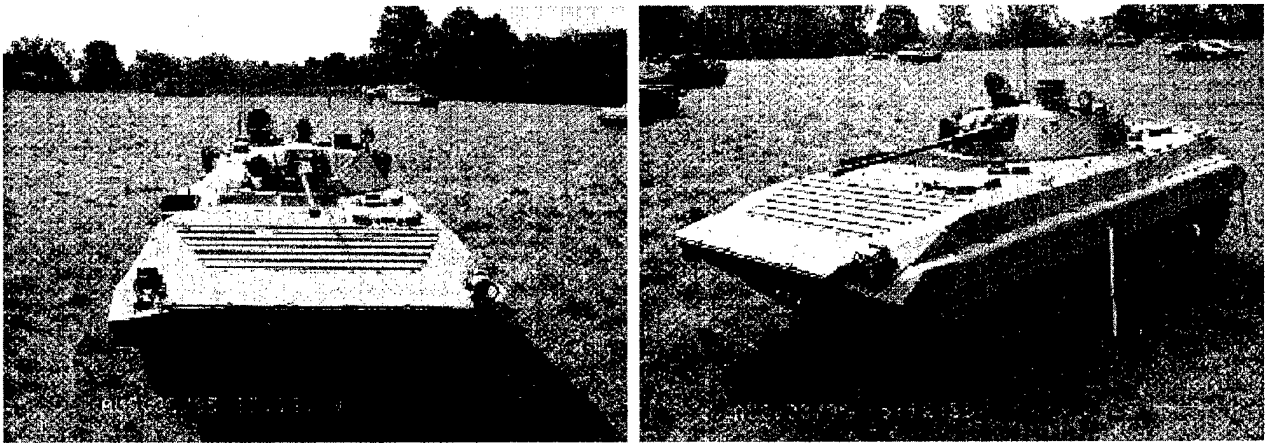

Figure 4.2: BMP2 infantry fighting vehicle in a C71 configuration.

algorithm's performance to be evaluated when the target data is subjected to significant intraclass SAR signature variability. Photographic images of one configuration of each of the three target vehicles are shown in Figures 4.1–4.3. The other configurations are shown in Appendix A.

The publicly released data was collected in September 1995 using an X-band radar operating at 9.6 GHz. The public release data was collected at two depression angles: $15^o$ and $17^o$ with a one foot resolution in both the range and cross range dimensions. The target data was collected with the radar in a spotlight mode, and the clutter data was collected with the radar in a strip map mode. A 35 dB Taylor weighting was applied in both the range and cross-range processing to reduce sidelobe levels. The target data was delivered in a "chip" format consisting of $128 \times 128$ pixels covering an area of approximately $26 \times 26$ meters. Each chip contained one target signature at one aspect angle. Figure 4.4 is an example of a T72 target chip at a zero degree aspect angle. The ground range represented by each pixel is 0.202 meters in range and 0.203 meters in cross range. Note the target and target induced shadow regions contained in this SAR image chip. SAR target images were collected over the

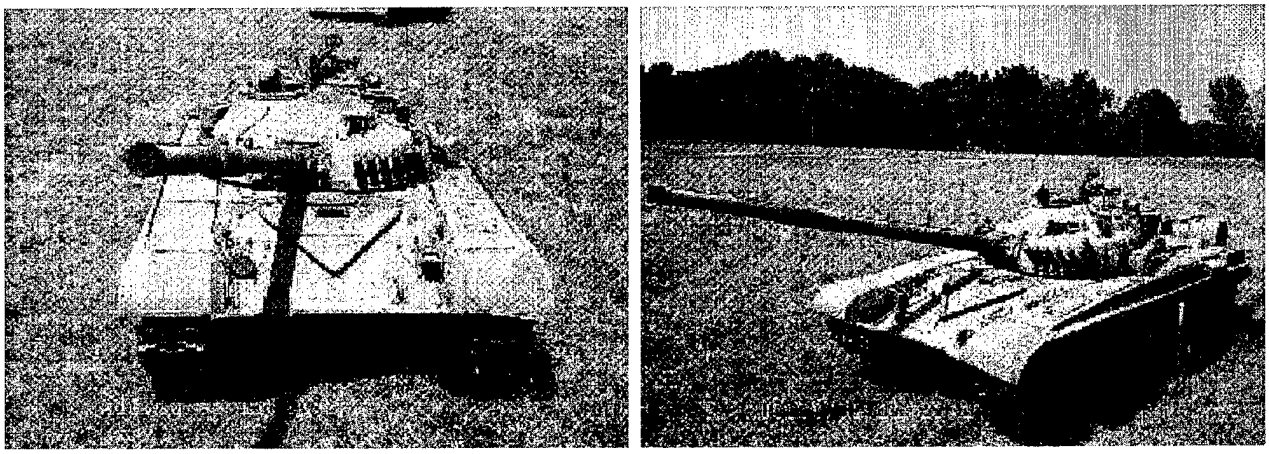

Figure 4.3: T72 tank in a 812 configuration.

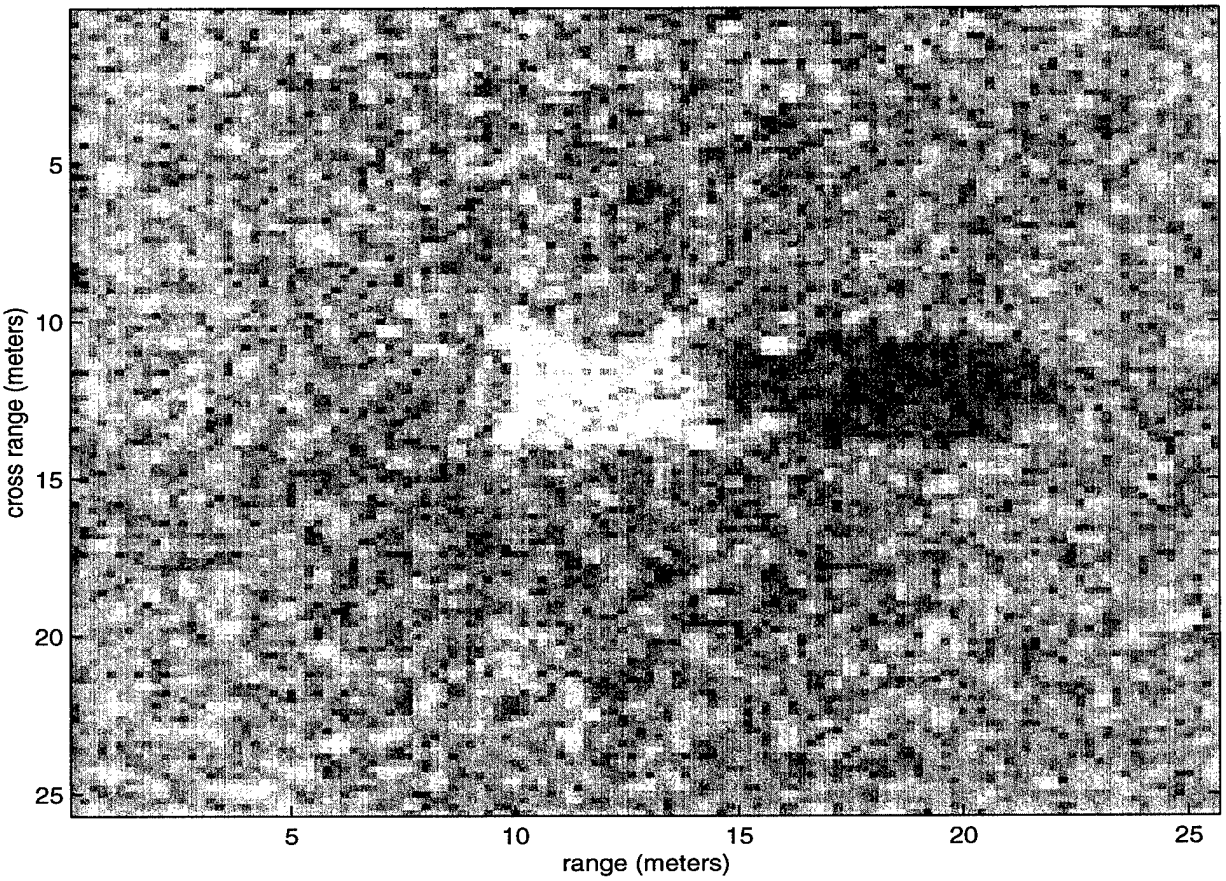


Figure 4.4: Example T72 tank SAR image a zero degree azimuth angle.

full target aspect range of $360^o$ at approximately $1^o$ increments for each target configuration.

### 4.1.2 MSTAR Clutter Images

The MSTAR clutter images consist of $1784 \times 1472$ pixels covering approximately 360 meters $\times 297$ meters. The ground range represented by each clutter pixel is also 0.202 meters in range and 0.203 meters in cross range. Appendix B shows example MSTAR clutter images.

## 4.2 DSSA Experiments

A three tiered approach is used in the DSSA detection/classification algorithm to reduce the number of computations. The approach consists of implementing a DSSA detector stage that operates on a low resolution version of the SAR data. The use of low resolution SAR data reduces the complexity of the DSSA detector and also reduces the complexity of the SAR image formation process. Although the use of low resolution data reduces complexity, it may also result in increased false alarms and less classification capability at the first stage. But the increased number of false alarms and the decreased classification capability at the first low resolution stage can be resolved in the two latter stages. The second DSSA classification stage operates on medium resolution SAR data. If the DSSA classifier is not able to render confident classification decisions using medium resolution SAR data, then a final DSSA classification stage operates on high resolution SAR data to reach final target recognition decisions.

The multiple tiered approach requires the use of progressive SAR image formation algorithms. Once SAR image regions of interest have been detected at the first stage, these regions are then processed with a progressive SAR image formation algorithm to obtain localized, higher resolution SAR images. This approach has the benefit that complex SAR image formation algorithms can be simplified by selective application to smaller subregions of the SAR phase history data.

### 4.2.1 Computing Environment

The experiments were performed on a UNIX Sun workstation and on a Windows NT personal computer. The Sun was a SPARCcenter 1000 with two 85-MHz SuperSPARC II processors and 192 megabytes of RAM memory. The personal computer has an Intel 166-MHz Pentium processor with 128 megabytes of RAM. Custom C-language programs, C-Shell scripts and MatLab scripts were written to perform the experiments.

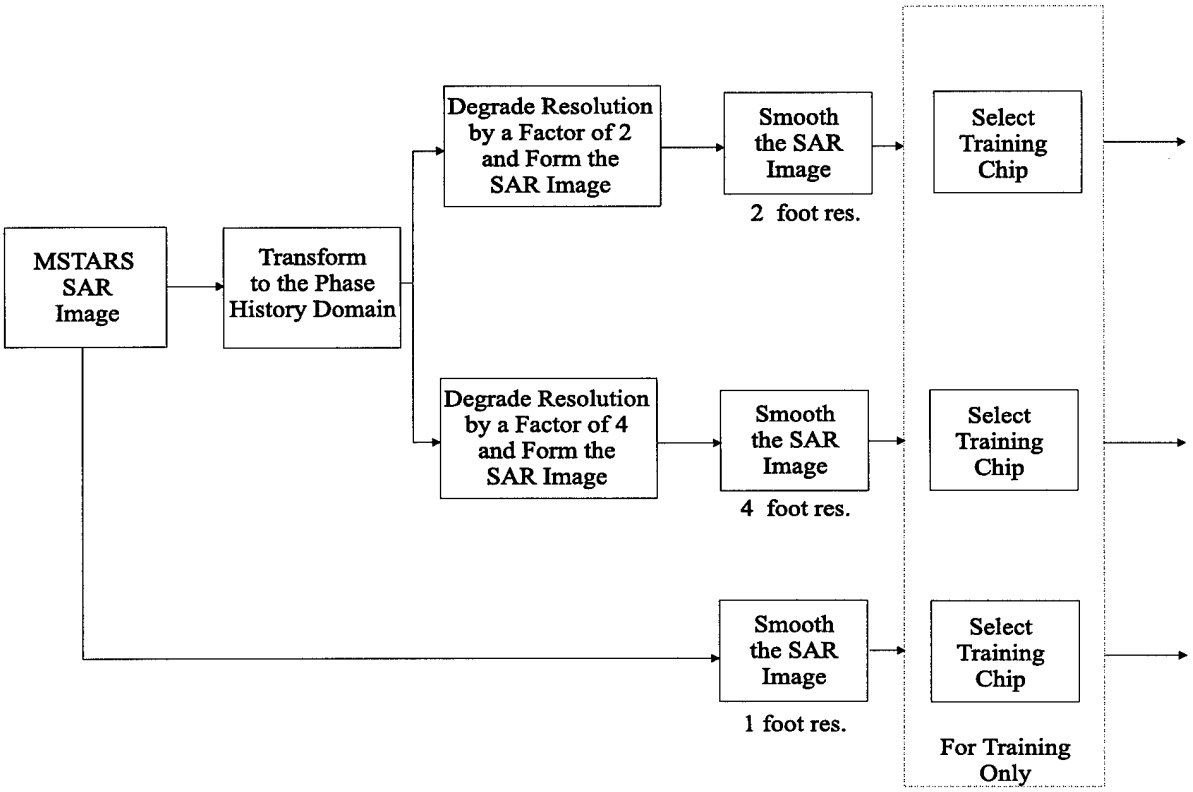


Figure 4.5: SAR resolution reduction algorithm.

### 4.2.2 Data Preprocessing

The DSSA experiments require SAR images at low, medium and high resolution levels. However, all MSTAR data is provided only at a high resolution level of 1 foot. Thus, it was necessary for GTRI to generate lower resolution versions of the MSTAR data sets. GTRI degraded original 1 foot resolution SAR data to 2 foot and 4 foot resolutions levels. Furthermore, the MSTAR data occasionally has data fall-out regions that are likely due to quantization errors in the radar's analog-to-digital converter. These fall-out regions can be recognized as exceptionally dark pixels in the SAR images. The MSTAR data was smoothed to reduce the impact of the fall-out regions. Smoothing also helps to reduce the speckle of the SAR imagery. The rest of this section describes the algorithm used by GTRI to generate low resolution versions of the MSTAR data, and the smoothing algorithm used to reduce the effects of fall-out and speckle.

#### SAR Resolution Reduction Processing

The process shown in Figure 4.5 was used to generate 2 and 4 foot resolution data from the high resolution 1 foot SAR data for both the target and clutter images. The SAR pixel data was first converted into phase history data. First, an inverse DFT and an inverse 35 dB Taylor weighting function is applied to the original SAR pixel data along the cross range

| Target Class | Number of Training Chips | Number of Testing Chips |
|---|---|---|
| T72 | 691 | 682 |
| BTR70 | 233 | 196 |
| BMP2 | 698 | 587 |

Table 4.1: Numbers of target images used for training and testing.

dimension, and then this is repeated in the range dimension to form SAR phase history data. Then, two data sets are formed from the resulting phase history. In one set the phase history is truncated by discarding half of the phase history samples in both dimensions. The second data set is obtained by discarding 3/4 of the phase history samples. This truncation of the phase history data results in a degradation of the resolution in both dimensions by a factor of 2 and 4, respectively. At this point, forward SAR signal processing is performed on these subapertures to form the lower resolution SAR images. A 35 dB Taylor weighting and DFT is applied to the truncated phase histories in the range dimension and cross range dimension to form the degraded resolution SAR images. A log function (base 10) was then applied to the SAR image data to compress the range of pixel values.

The high resolution SAR clutter images contain $1472 \times 1784$ pixels, the medium resolution SAR clutter images contain $736 \times 892$ pixels, and the low resolution SAR clutter images contain $368 \times 446$ pixels. Examples of the clutter images at different resolutions are given in Figures 4.6—4.8.

#### Smoothing

A sliding window average is applied to the SAR images to reduce the effects of fall-out and the speckle associated with the SAR images. The sliding window is sized to contain a pixel and its eight nearest neighbors as shown in Figure 4.9. An average of the nine pixel values is then used to replace the value associated with the center pixel. Image edge boundaries are accounted for by reducing the size of the window. The resolution reduction described in the previous section and this smoothing process were performed on all of the target and clutter images.

### 4.2.3 DSSA Detector/Classifier Training

#### DSSA Training Sets

The target data collected at a $17^o$ depression angle was used for training the DSSA algorithms, and the clutter images and target images collected at a $15^o$ depression angle were used for testing. The numbers of target chips used for training and testing are given in Table 4.1 for each composite (all configurations) target class. The number of chips is a factor of

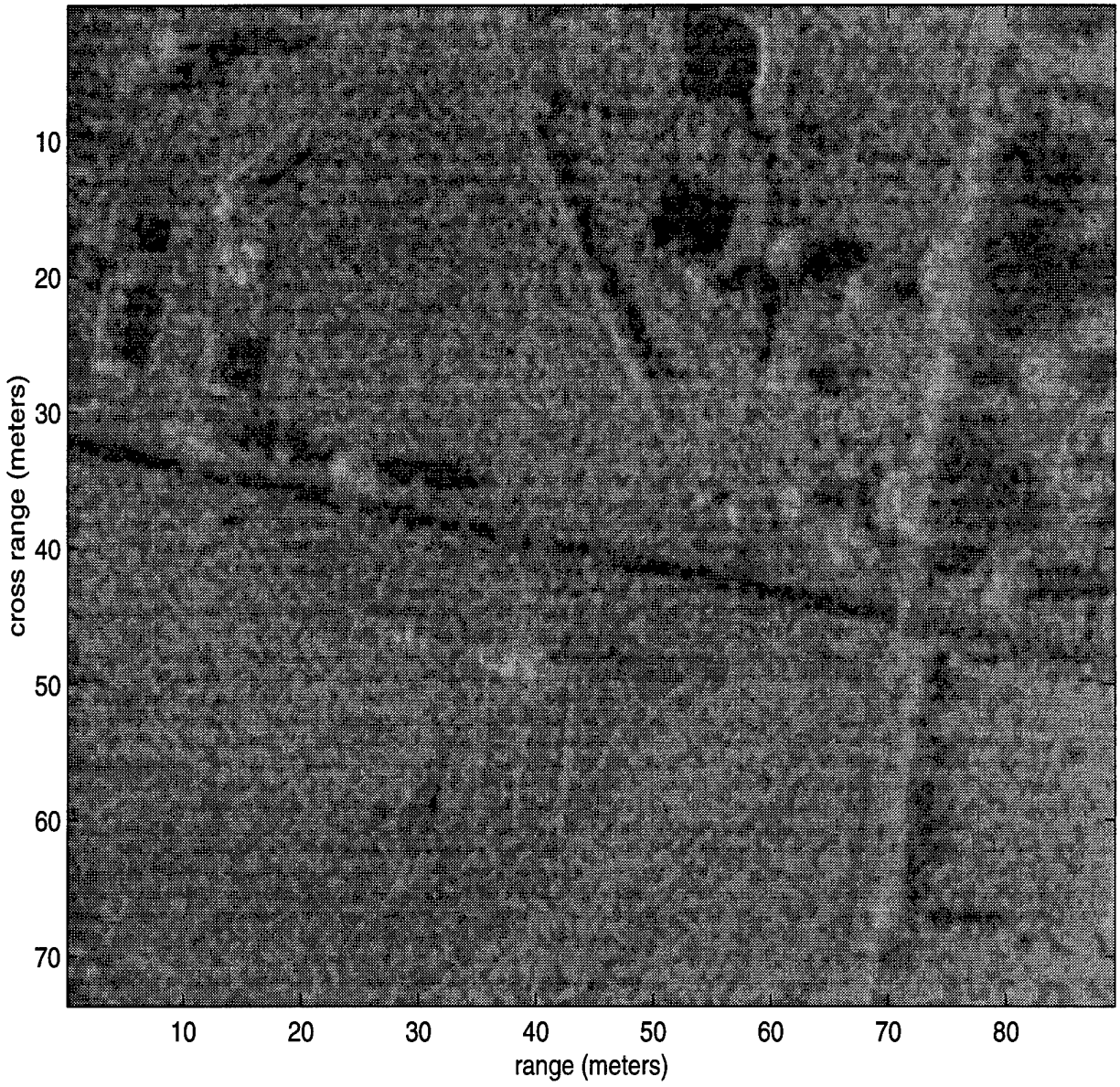


Figure 4.6: Example SAR clutter scene at low resolution.

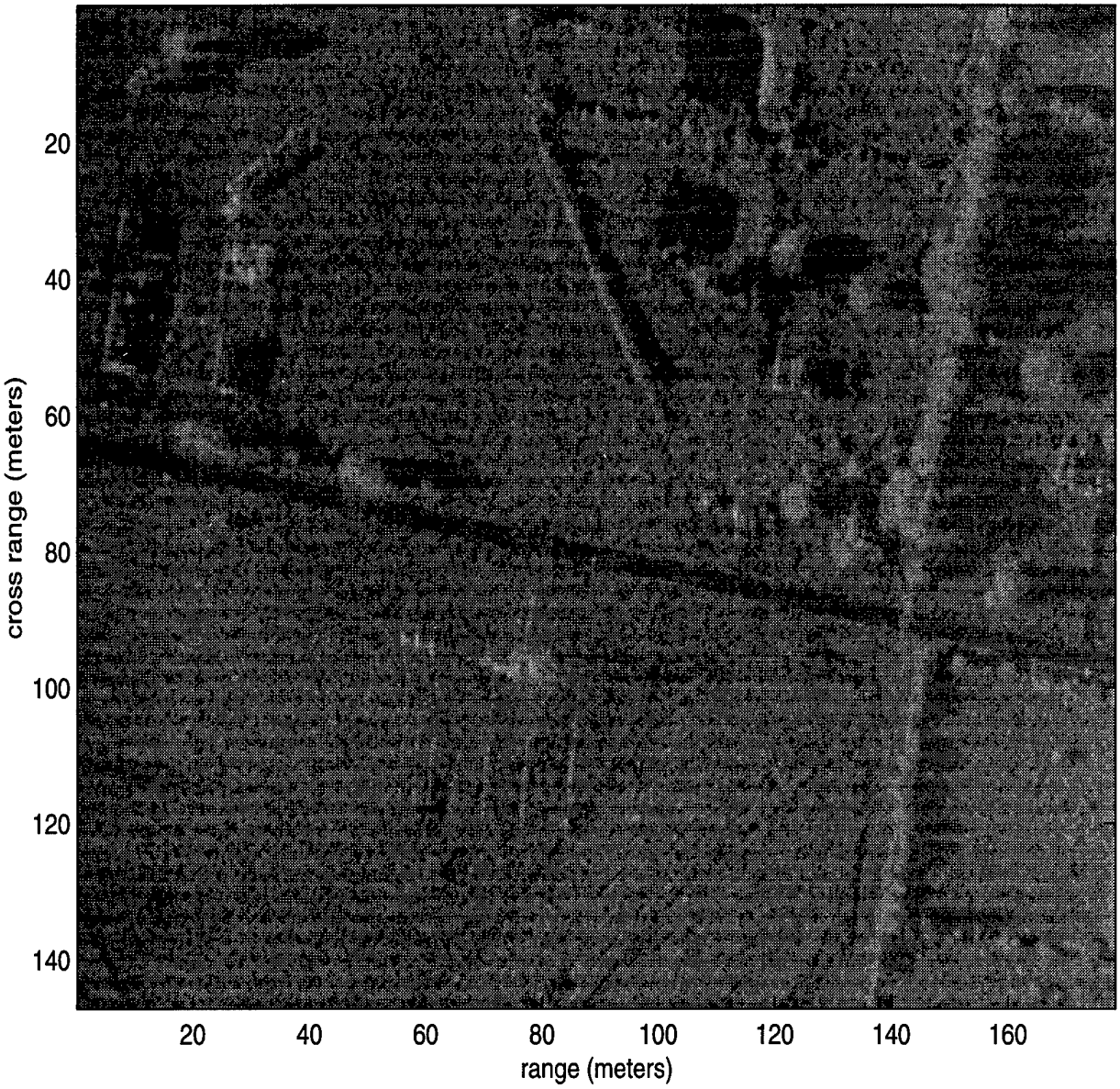


Figure 4.7: Example SAR clutter scene at medium resolution.

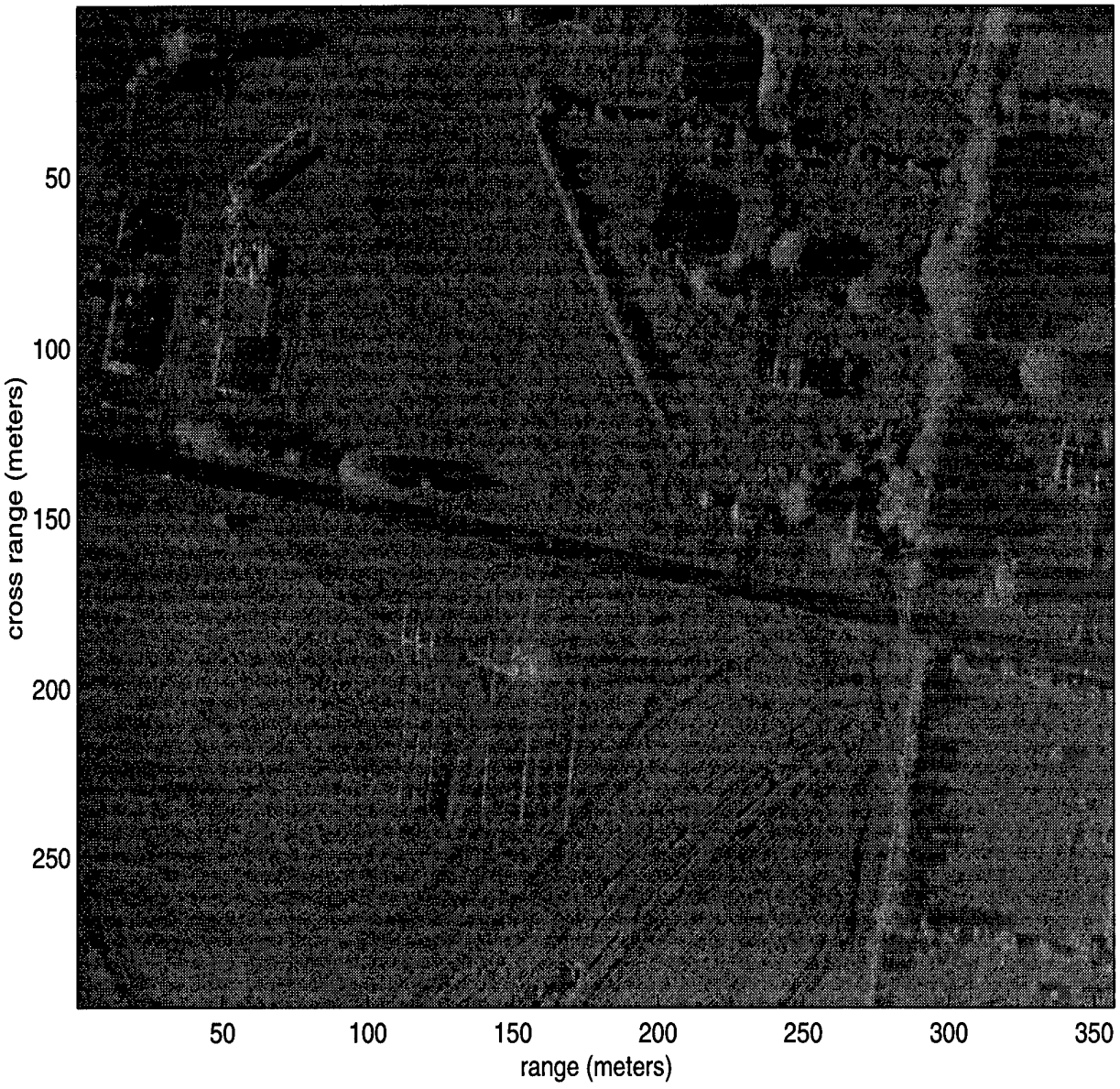


Figure 4.8: Example SAR clutter scene at high resolution.

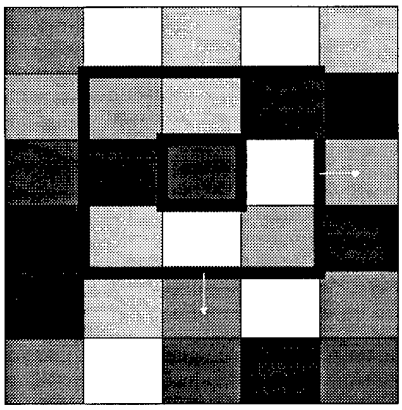

Figure 4.9: Sliding window used for SAR image smoothing.

| Resolution Level | Snippet Size | Number of Pixels |
|---|---|---|
| Low | $11 \times 17$ | 187 |
| Medium | $21 \times 33$ | 693 |
| High | $41 \times 65$ | 2665 |

Table 4.2: Snippet sizes extracted from target chips.

three larger for the T72 and BMP2 target classes due to the three baseline configurations included for these targets. The data from the different configurations were mixed together to form training and testing data for each target.

### DSSA Feature Vectors Definition

Each target chip consists of $128 \times 128$ pixels at the one foot resolution and $64 \times 64$ and $32 \times 32$ pixels at the two and four foot resolutions, respectively. However, the $26 \times 26$ meter target chips contain large regions void of target returns or target shadow. Therefore, in defining a DSSA "snippet" size to use for extracting pixels for insertion into the DSSA feature vectors, subregions of the target chips were selected to isolated the target signatures. These snippets contain mostly target returns and target shadows. The sizes of these subregions for each resolution level are given in Table 4.2. Sample snippets extracted for one of the T72 target chip series are shown in Figure 4.10 for the different resolutions (note the fall-out region in the high-resolution snippet).

### DSSA Template Generation

DSSA templates sets where generated for each target training set. The number of DSSA stages varied from 8 to 16 stages, depending upon the particular target class and resolution level. Each DSSA stage contained 32 templates for the tank and fighting vehicle classes, and each DSSA stage contained 16 templates for the transport vehicle target class. These

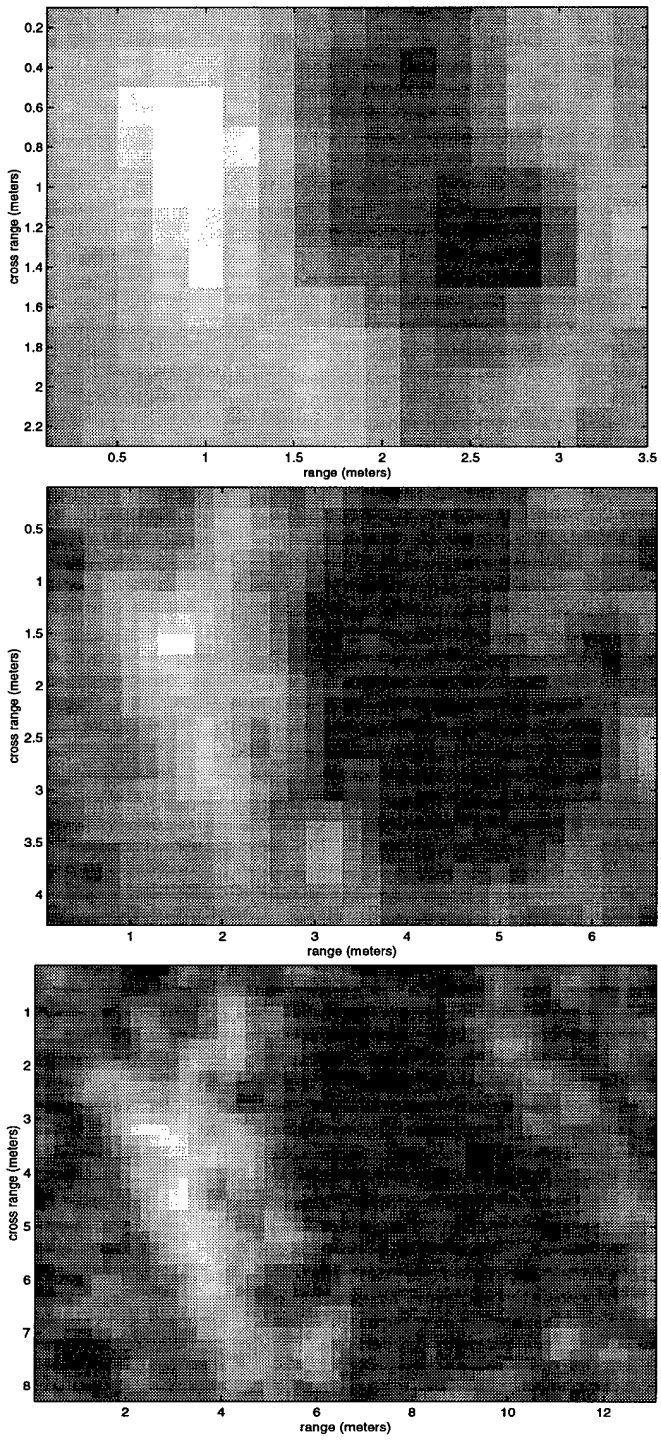


Figure 4.10: Example target snippets used to form DSSA feature vectors.

different numbers of templates is a result of different configuration data being used to form different amounts of training data.

### 4.2.4 DSSA Detector/Classifier Testing

The DSSA detector/classifier processes the SAR imagery in a progressive manner from low resolution to high resolution SAR. Only those low resolution regions-of-interest that pass the DSSA detector are subsequently processed to form (localized) medium resolution SAR imagery. These medium resolution subimages are then processed by the DSSA classifier. If a confident classification decision is reached with the medium resolution DSSA classifier, then the process terminates, otherwise, additional SAR image formation processing is performed to form high resolution versions of each suspect region of interest. The final DSSA classifier then uses the high resolution SAR data to make a final decision. However, remember that there is no architectural or basic operational differences between the DSSA detector and DSSA classifier; only the size of the feature vectors and required ($SNR_{mm}$) confidence levels differ.

Here, "confidence" is simply quantified as the level of mean-squared-error (MSE) between the DSSA input data and the DSSA templates. If the MSE is low, confidence is high. Thresholds are associated with each DSSA template in a decision tree to judge the quality of its associated MSE for a given input feature vector.

A sliding window is used to extract a snippet from each pixel location in the low resolution SAR image to form DSSA feature vectors. Thus, the DSSA detector/classifier returns a MSE value for each pixel location in the processed image. This MSE-based template dissimilarity map is then subsequently processed to reduce the rate of false alarms with simple constant false alarm rate (CFAR) algorithms described in the next section.

#### CFAR Postprocessing

Post-processing software was developed that converts the MSE distance measures acquired from each target-specific DSSA detector/classifier into into a signal-to-mismatch noise ratio ($SNR_{mm}$). The variance of the snippet contained in the original SAR data at each pixel location (signal energy) is normalized by the measured MSE value (template mismatch noise). This signal-to-noise ratio provides a similarity surface that is subsequently processed with a set of simple CFAR algorithms.

The simple CFAR algorithms threshold the peaks of the ($SNR_{mm}$) surface, and count the number of pixels within each peak at the given (possibly multiple) threshold levels. If the pixel counts are within acceptable ranges, a target detection is declared. In this research, these pixel count ranges were optimized to give the best possible performance for the MSTAR target test images. The MSTAR clutter images were not used in choosing these CFAR parameters.

| High Resolution SAR | | | | |
|---|---|---|---|---|
| True | DSSA Decisions | | | False Alarms/Image |
| | BTR-70 | T72 | BMP-2 | |
| BTR-70 | 188 | 4 | 1 | 0.00 |
| T72 | 0 | 582 | 0 | 0.00 |
| BMP-2 | 0 | 12 | 574 | 0.00 |

| Medium Resolution SAR | | | | |
|---|---|---|---|---|
| True | DSSA Decisions | | | False Alarms/Image |
| | BTR-70 | T72 | BMP-2 | |
| BTR-70 | 172 | 10 | 11 | 0.00 |
| T72 | 0 | 572 | 10 | 0.07 |
| BMP-2 | 1 | 63 | 522 | 0.09 |

| Low Resolution SAR | | | | |
|---|---|---|---|---|
| True | DSSA Decisions | | | False Alarms/Image |
| | BTR-70 | T72 | BMP-2 | |
| BTR-70 | 121 | 37 | 35 | 1.50 |
| T72 | 19 | 488 | 75 | 1.24 |
| BMP-2 | 58 | 191 | 33 | 2.33 |

Table 4.3: Classification confusion matrices.

#### DSSA Classification Decisions

The DSSA classification rule is to simply pick the class of the DSSA classifier that provides the best DSSA template match.

### 4.2.5 DSSA Detection/Classification Results

A total of 1365 targets were contained in the test set. The first stage DSSA detector found 1361 of these in the low resolution SAR images, giving a detection rate of 99.70%. An average of five false alarms per image occurred at the low resolution level. Of the 1361 targets detected at the low resolution, the DSSA classifiers correctly classified 1361 at the medium and/or high resolution levels, giving a correct classification rate of 98.75%. An average of one false alarm in six images occurred at the medium resolution, and no false alarms were detected in the 100 test images at the highest resolution level.

Confusion matrices for the low, medium and high resolution levels are given in Table 4.3. Each row shows the number of correct and incorrect DSSA decisions for a single target class. Most errors at the high resolution level involved misclassifying the fighting vehicle, with its mounted machine gun, as a tank. One hundred SAR images containing rural and suburban clutter scenes were also tested at the low resolution level to estimate the false alarm rate.

Each image covered about one-tenth of a square kilometer.

Figure 4.14 and its transparency overlays show an example DSSA detection/classification result for a low resolution clutter image that also contains snippets of each of the three target classes. These test targets were inserted by hand by GTRI (since no clutter images with targets were provided by DARPA) to illustrate DSSA detector/classifier operation. The three overlays show the DSSA detection results generated by the DSSA template sets for each of the target classes. The first overlay is from the T72 templates, the second from the BTR70 templates, and the third from the BMP2 templates. Note the three false alarms collectively produced by the DSSA detectors. Also note that the overly color code indicates confidence by showing $\text{SNR}_{mm}$ values. Note that even the DSSA-detector has some ability to discriminate between the target classes even with low resolution imagery.

Figure 4.19, with its transparency overlays, shows an example DSSA detection/classification result for the same image, but at a medium resolution level. Only those regions that were detected at the low resolution level are processed at this level (for the T72 example, see Figure 4.15). Note that there are no false alarms. Also note that the color-coded $\text{SNR}_{mm}$ values show that the T72 has been classified at this medium resolution level with high confidence, but that there is still some uncertainty for the other two targets.

Figure 4.24 and its transparency overlays show an example DSSA detection/classification result for the same image, but at a high resolution level. Only those regions that were still active at the medium resolution level are processed at this level (for the T72 example, see Figure 4.20). The color-coded $\text{SNR}_{mm}$ values show that all targets are classified with significant confidence.

## 4.3 DSSA Implementation Costs

Although full implementation complexity and cost are difficult to quantify, the following sections provide memory cell and computational operation count analysis and measurements for the DSSA classifier and progressive SAR signal processing.

### 4.3.1 Memory Complexity

#### DSSA Basis Function Memory Requirements

The memory required for DSSA template storage is:

$$4k \times N \times P \text{ bytes} \tag{4.1}$$

where the DSSA basis functions are stored in single precision floating-point format (4 bytes per element), $k$ is the dimension of the feature space, $N$ is the number of DSSA basis functions per stage, and $P$ is the number of stages.

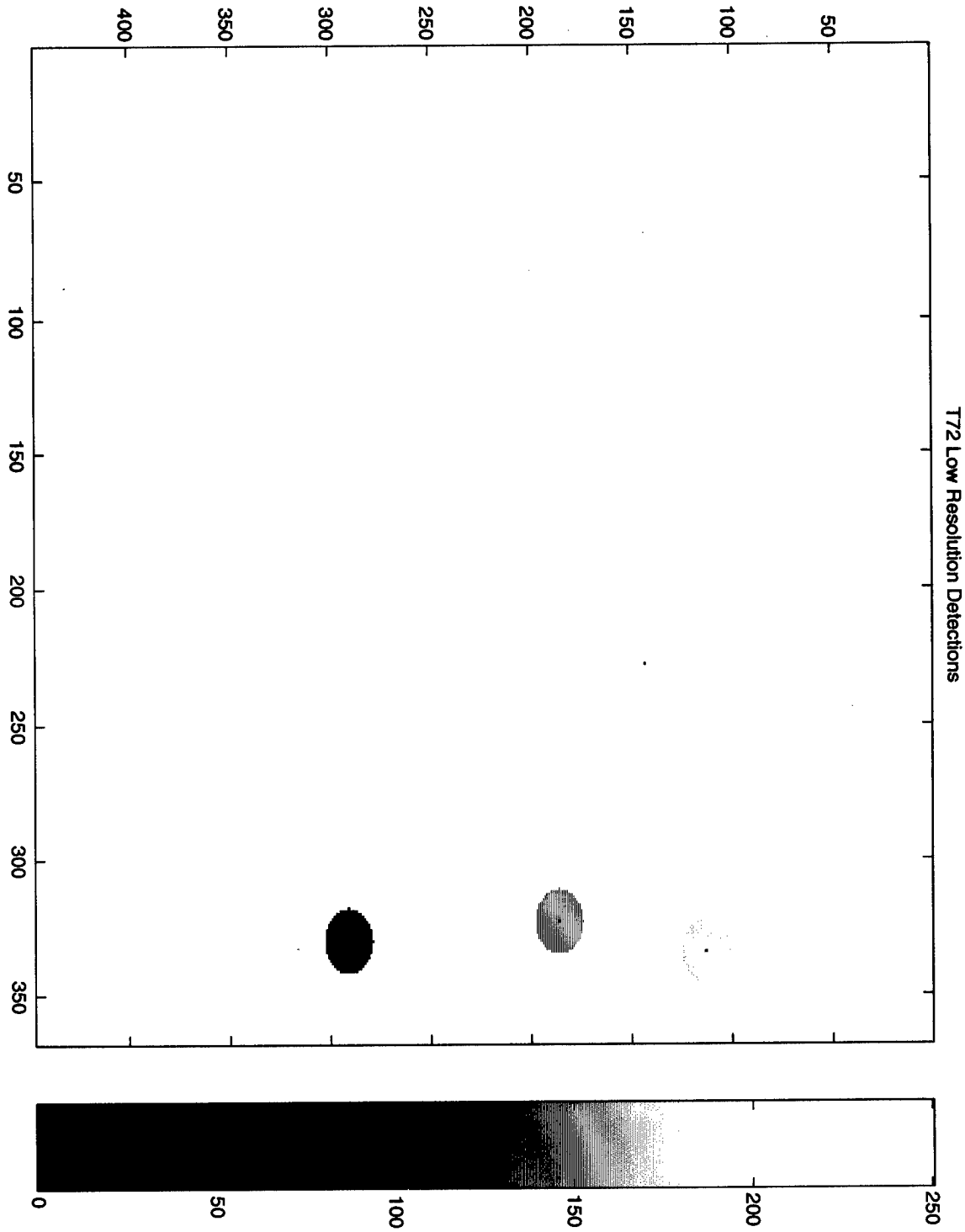


Figure 4.11: T72 detections for low resolution image.

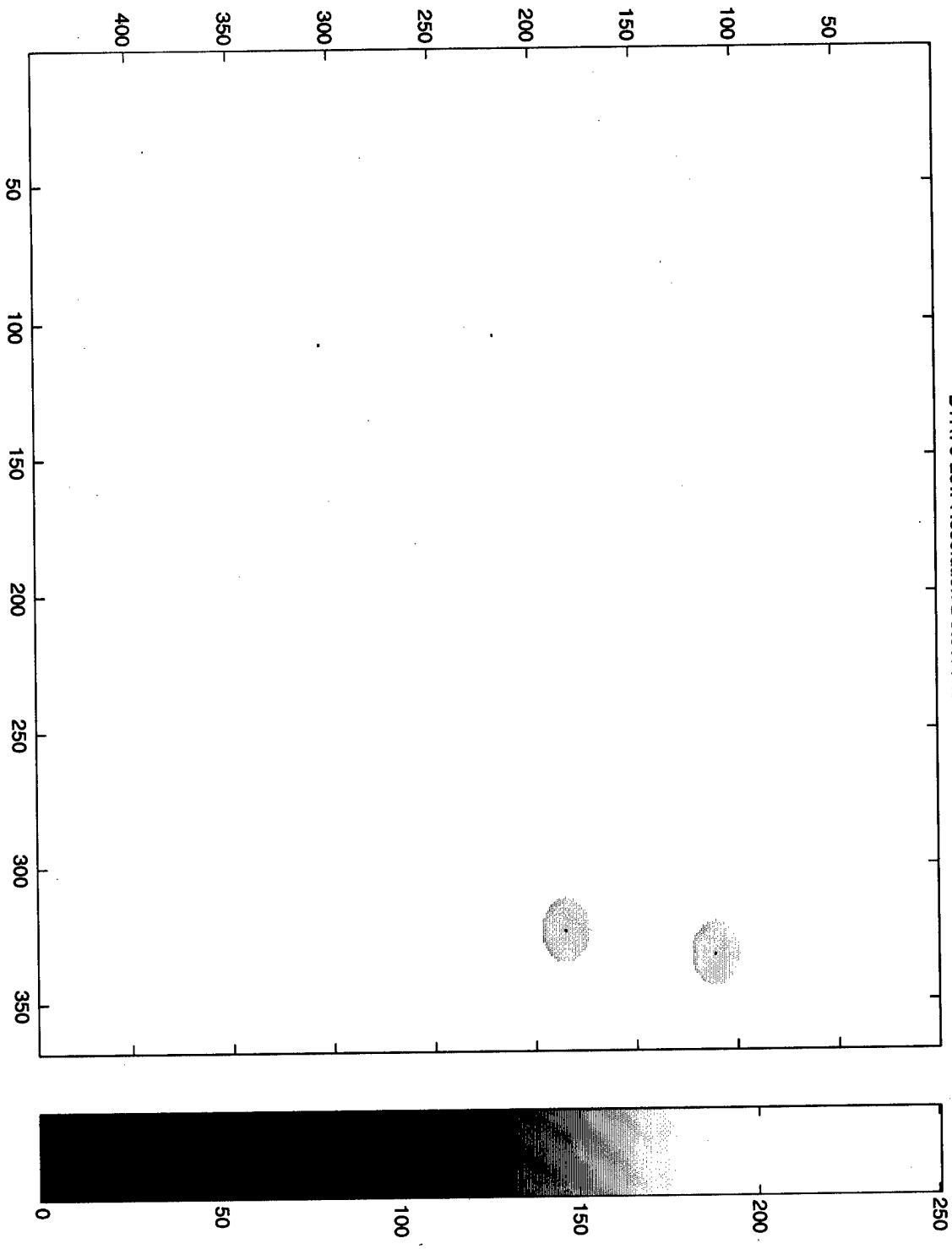


Figure 4.12: BTR70 detections for low resolution image.

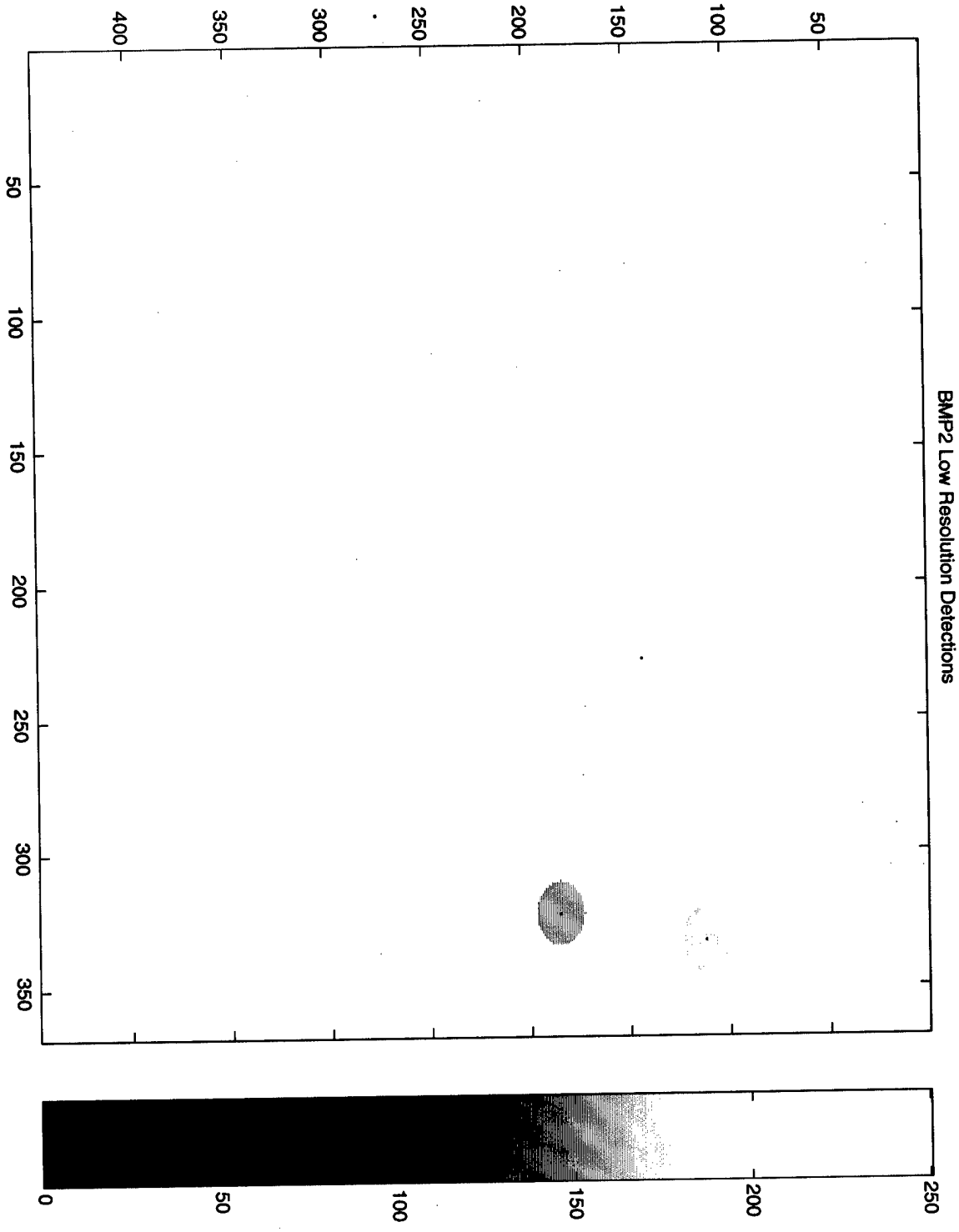


Figure 4.13: BMP2 detections for low resolution image.

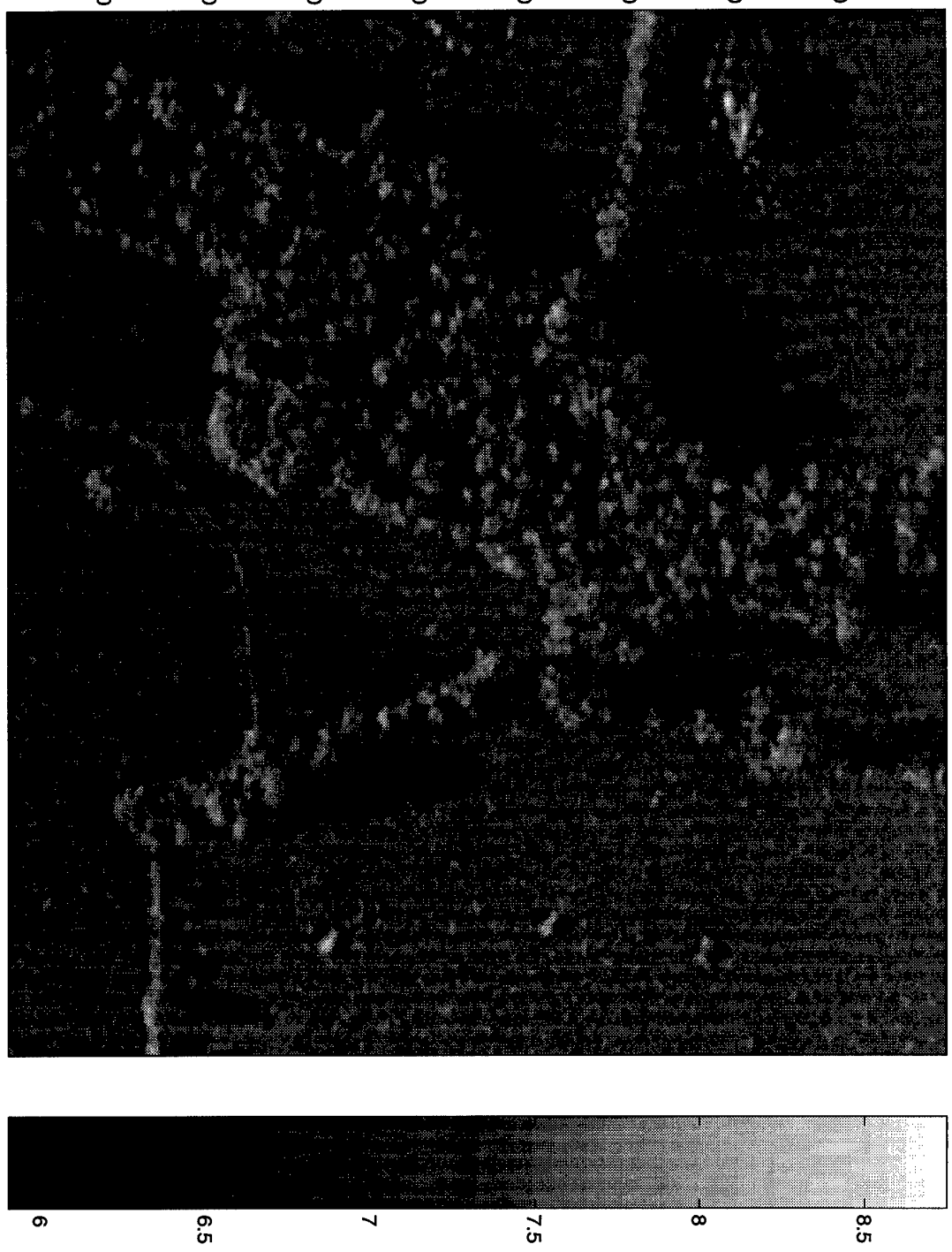


Figure 4.14: Example low resolution SAR test image.

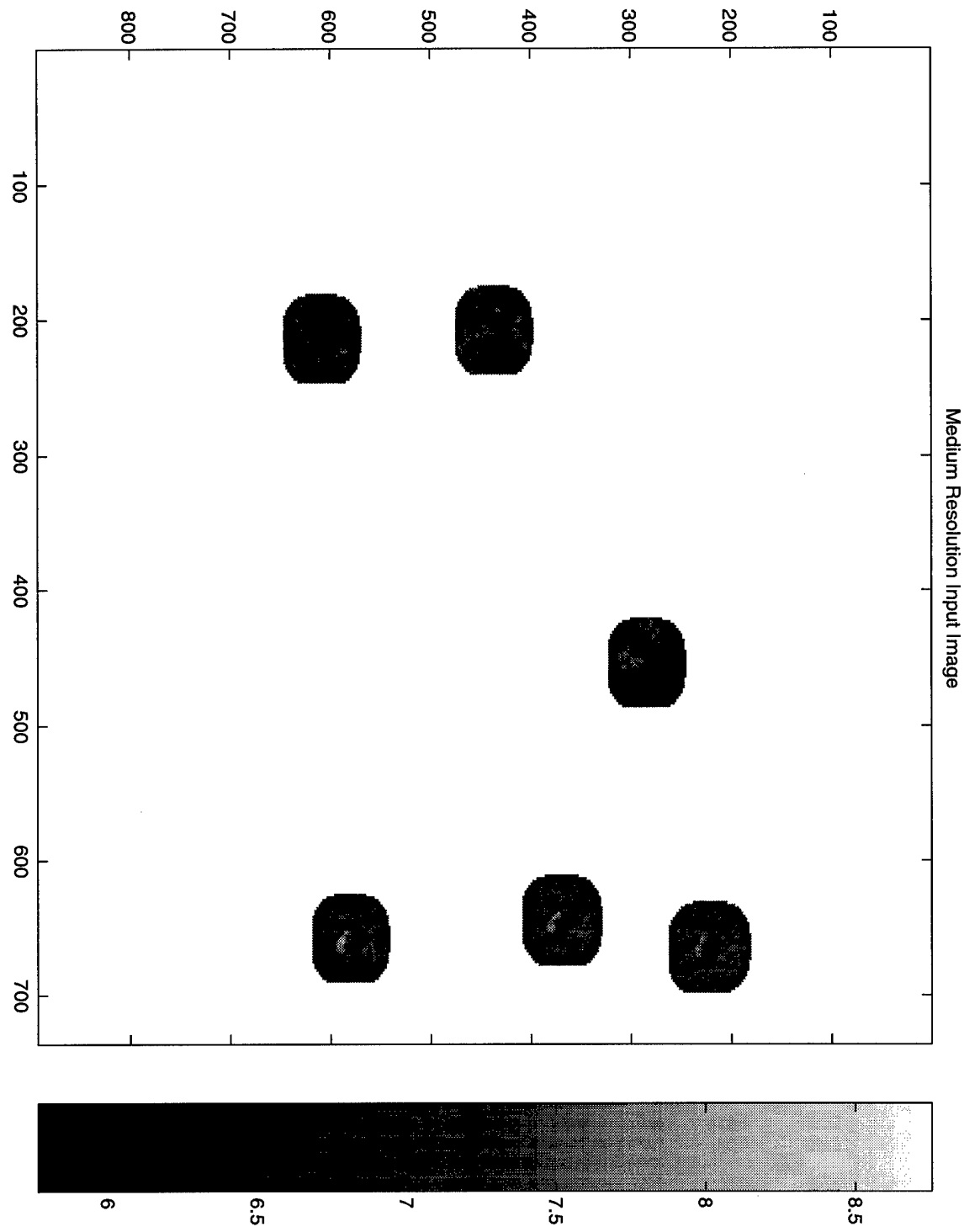


Figure 4.15: Identified regions-of-interest in medium resolution image.

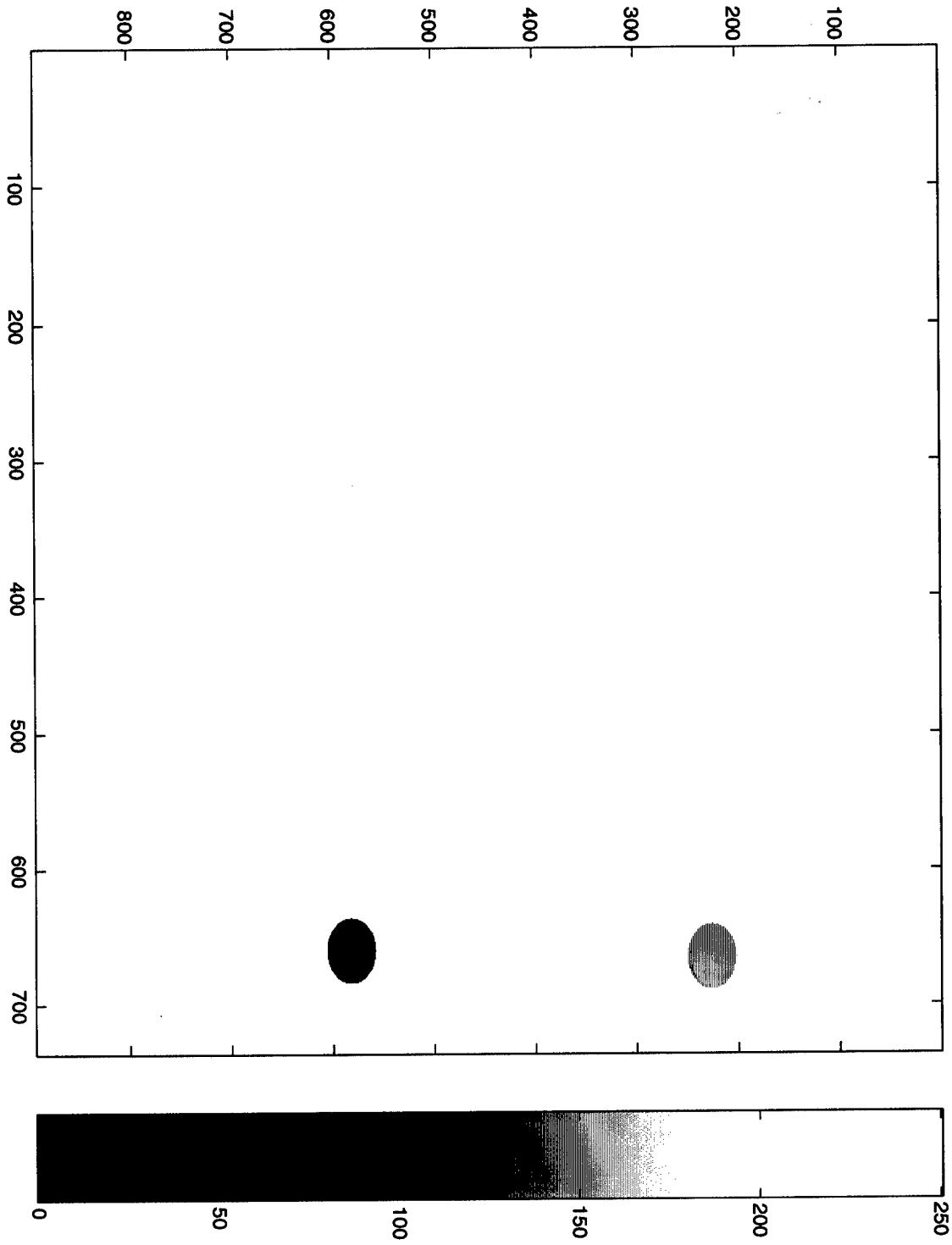


Figure 4.16: T72 detections for medium resolution image.

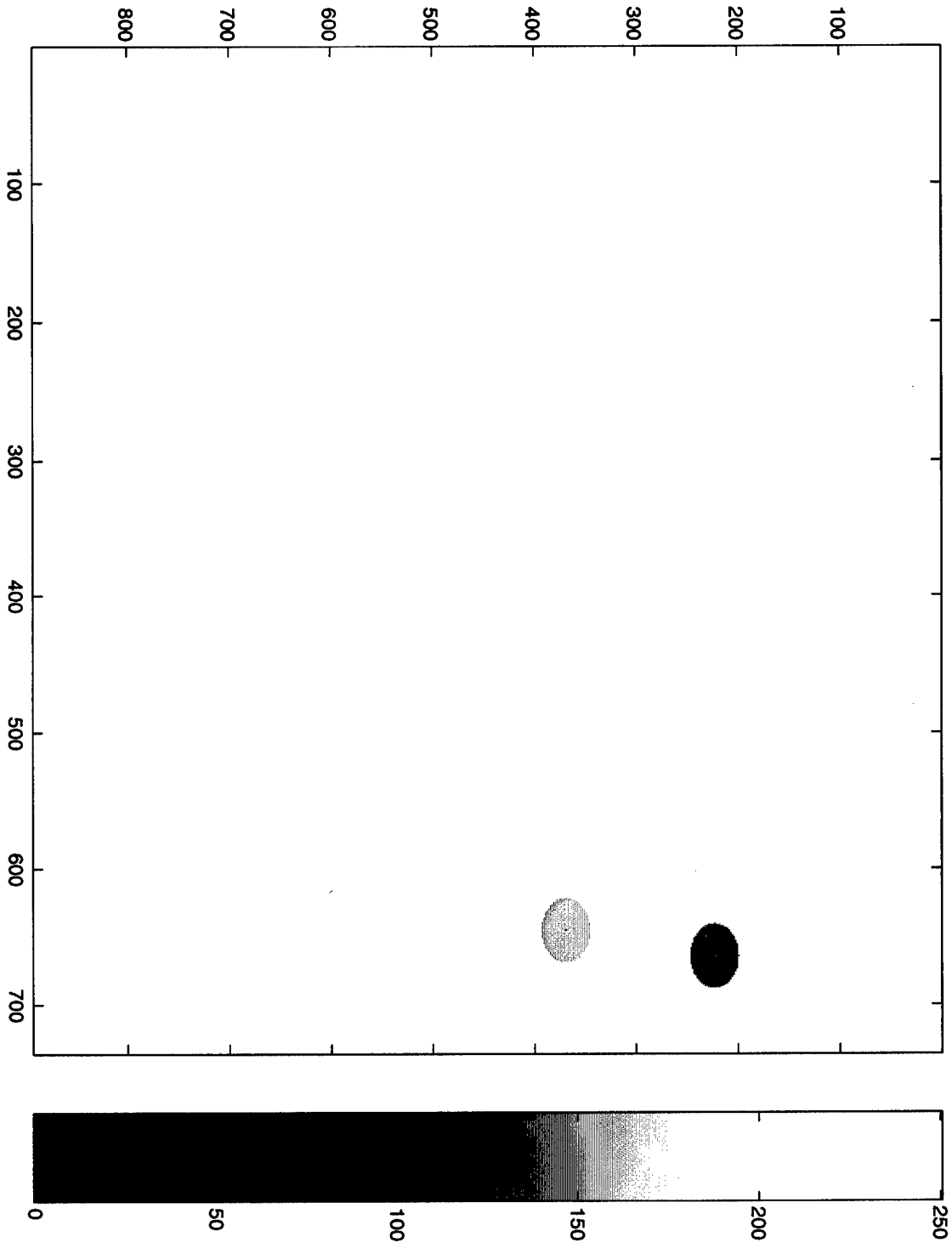


Figure 4.17: BTR70 detections for medium resolution image.

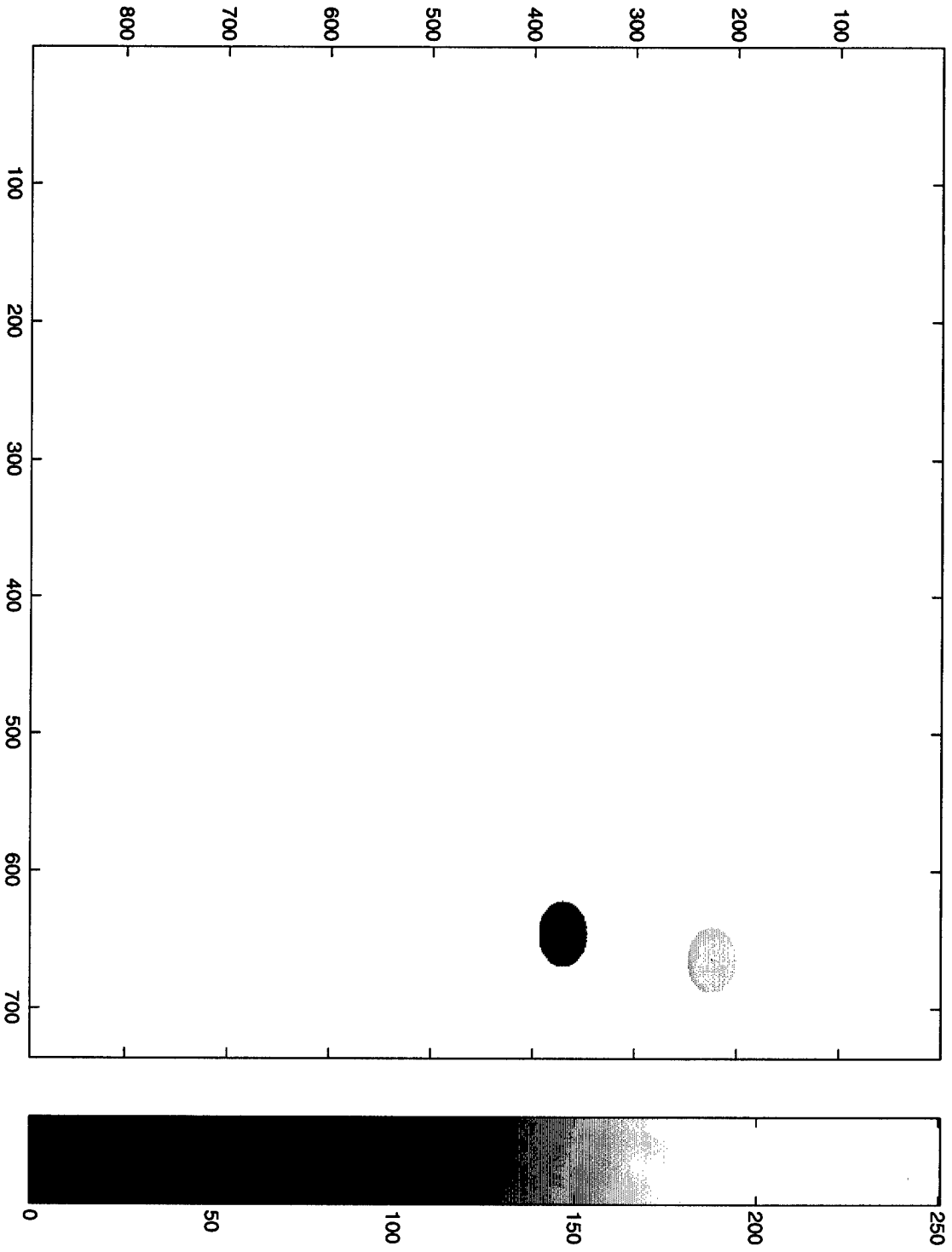


Figure 4.18: BMP2 detections for medium resolution image.

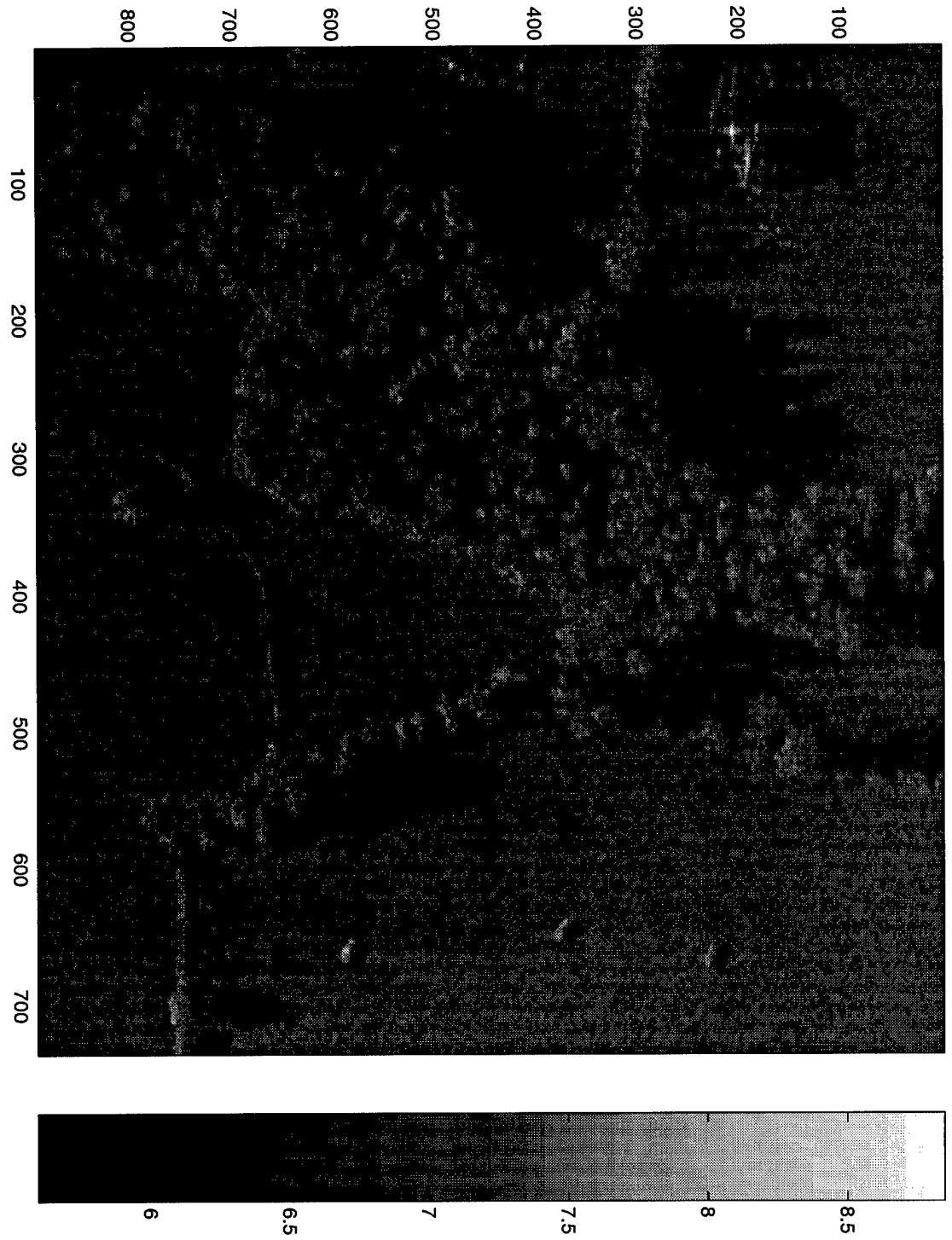


Figure 4.19: Example medium resolution SAR test image.

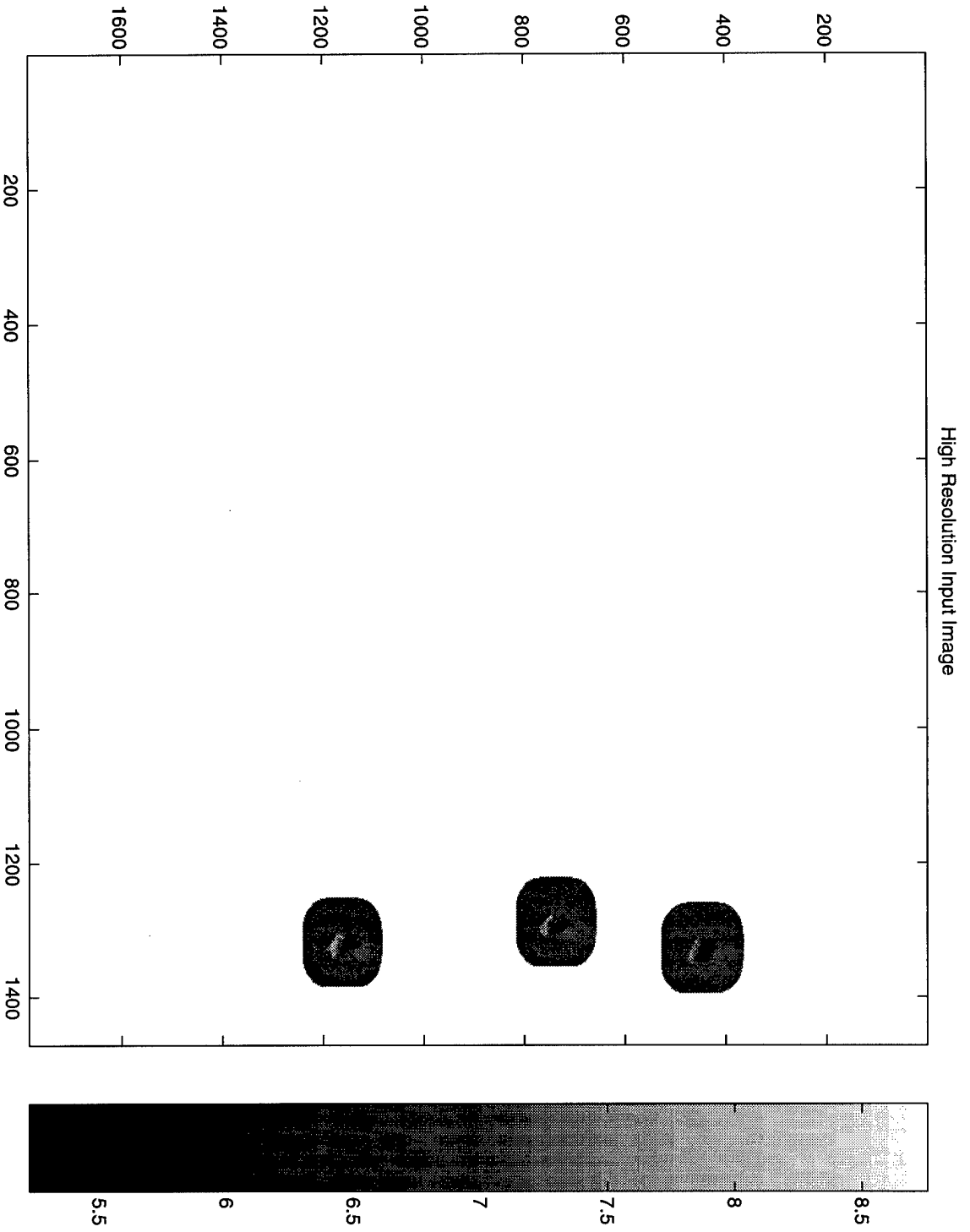


Figure 4.20: Identified regions-of-interest in high resolution image.

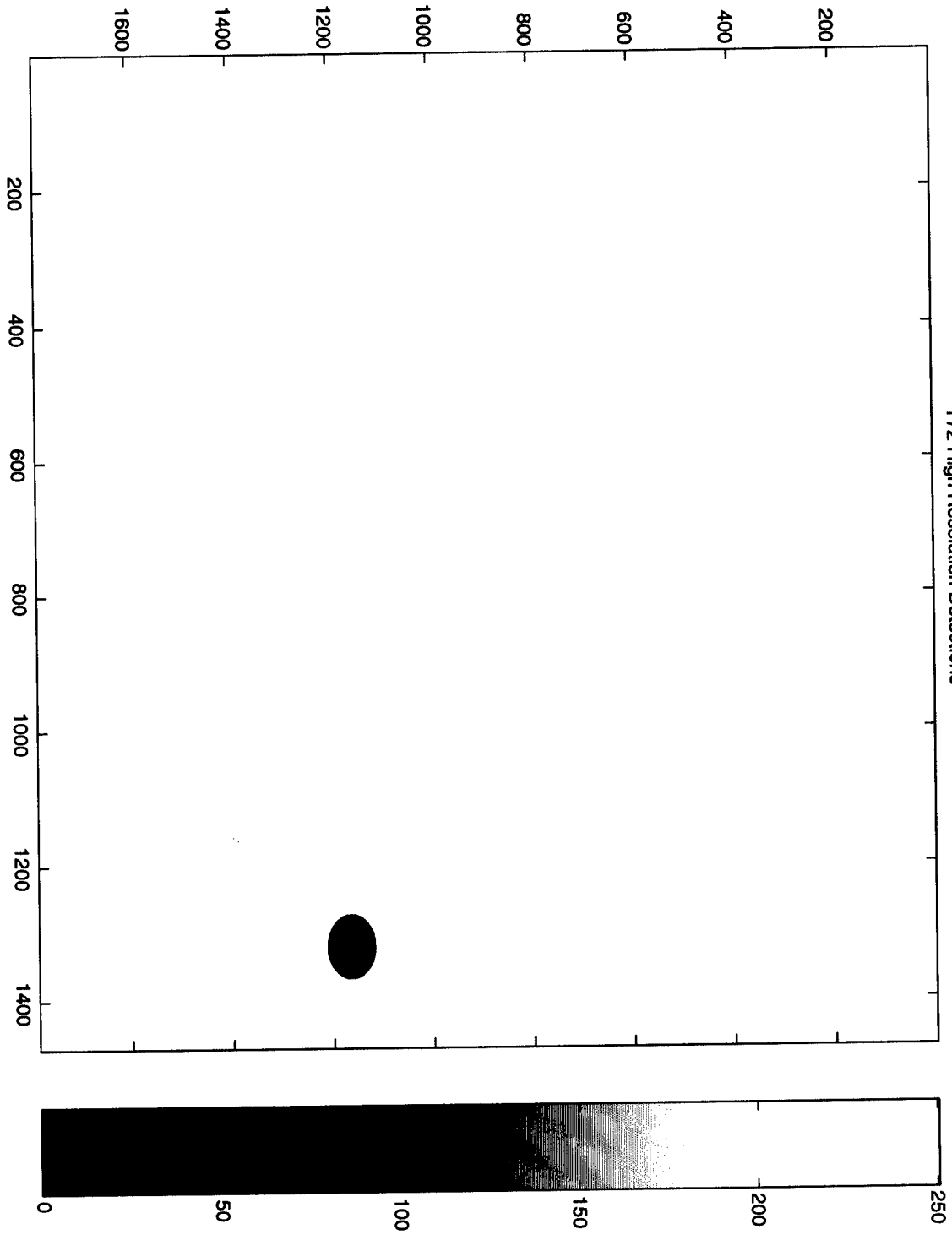


Figure 4.21: T72 detections for high resolution image.

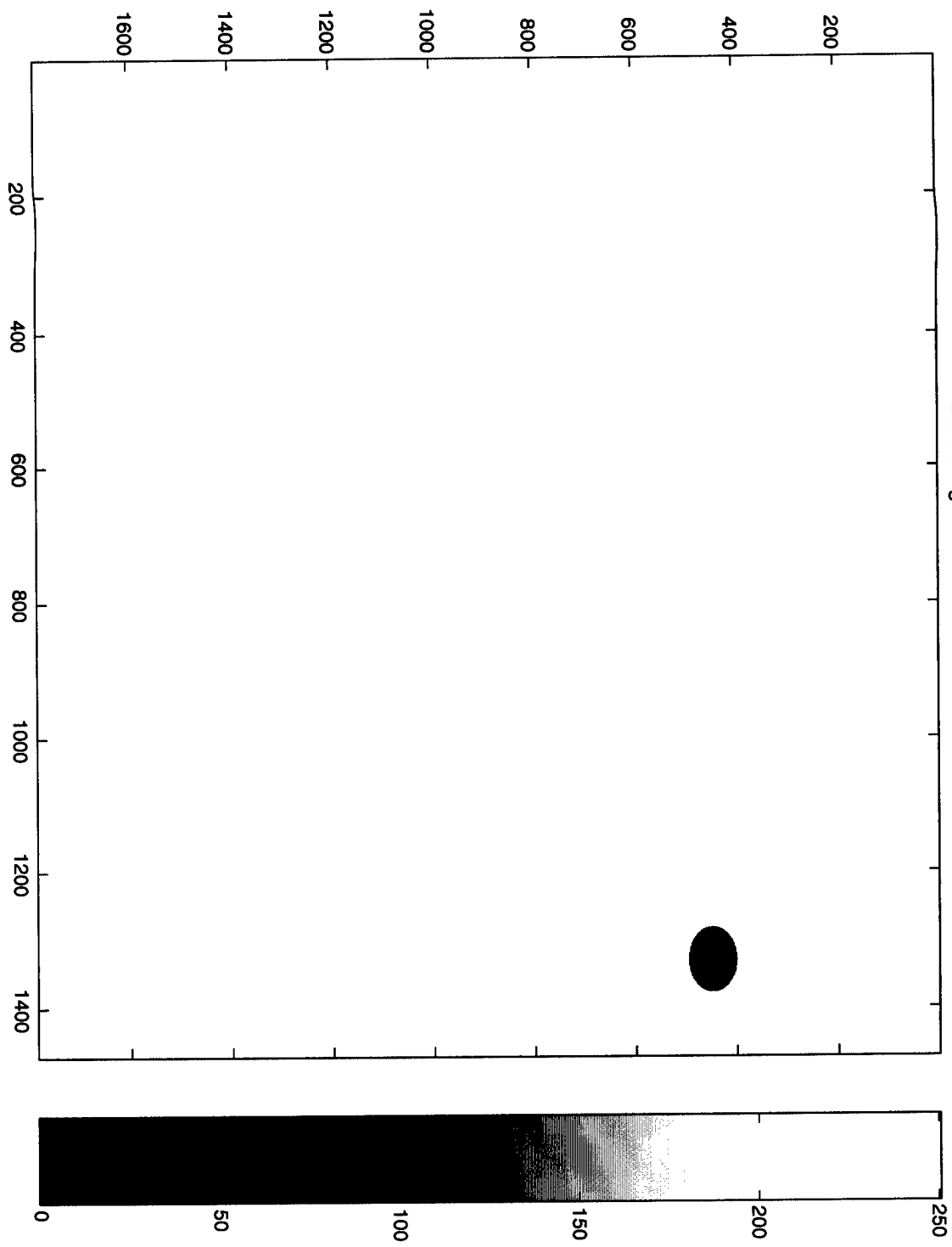


Figure 4.22: BTR70 detections for high resolution image.

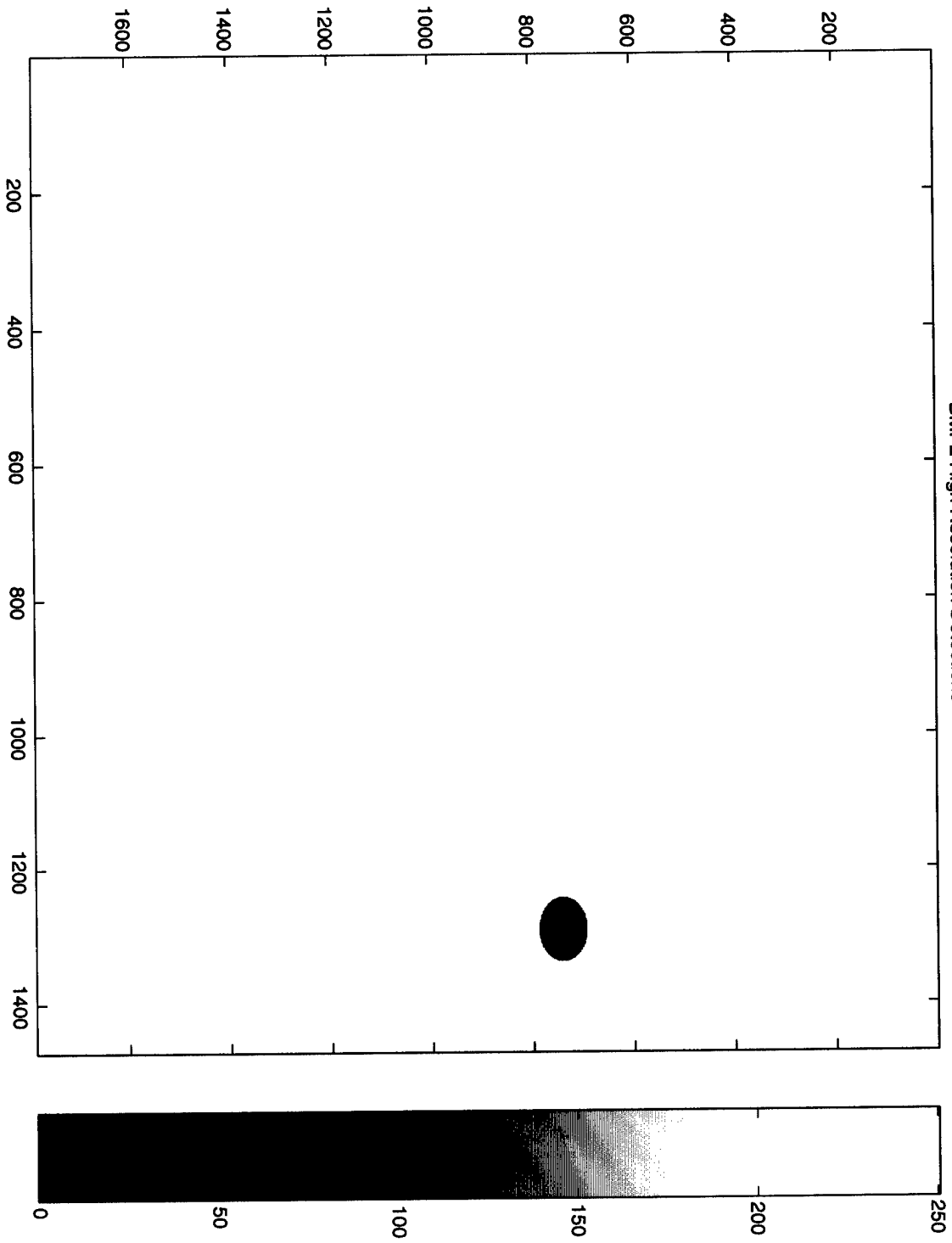


Figure 4.23: BMP2 detections for high resolution image.

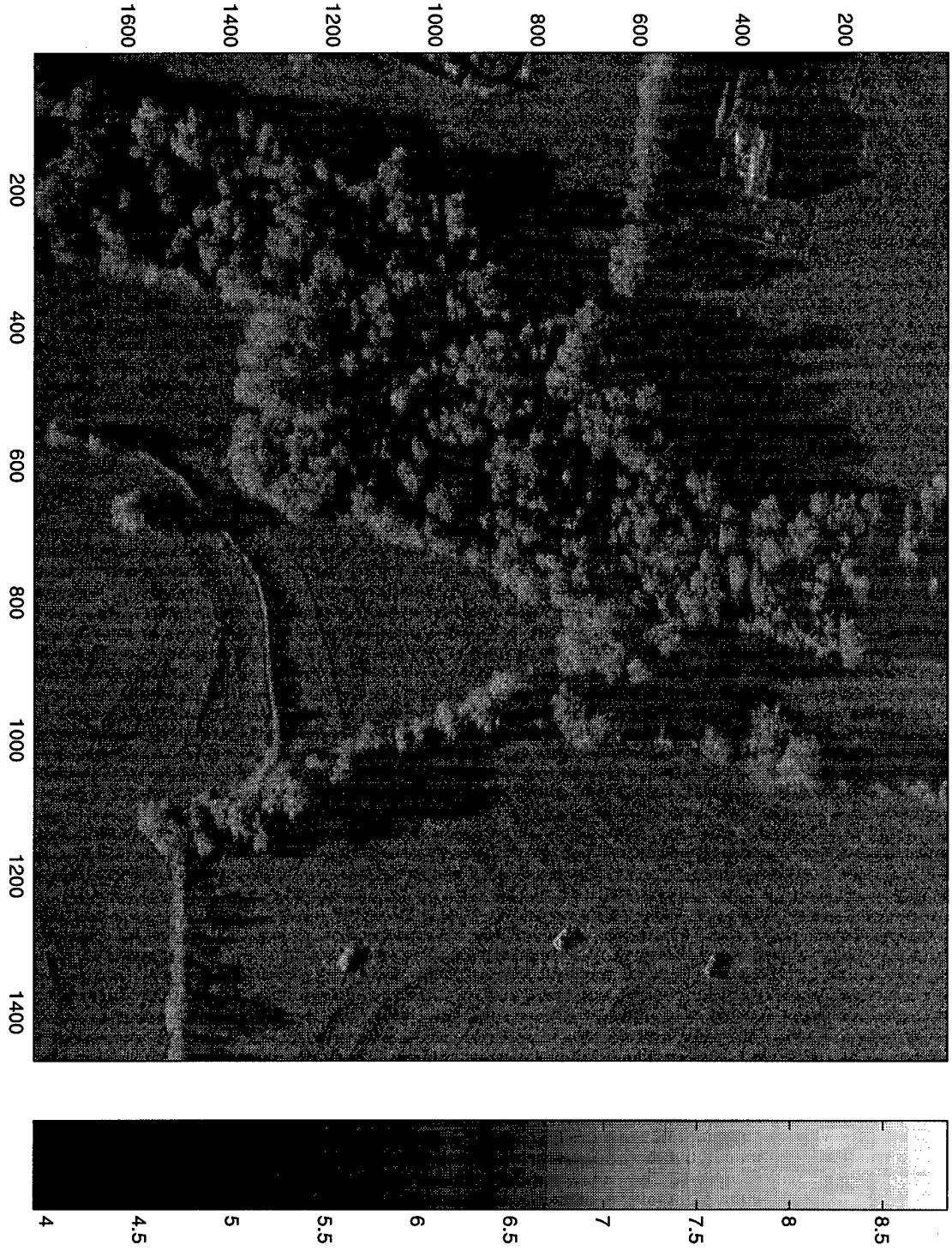


Figure 4.24: Example high resolution SAR test image.

| Feature Vector Size | Number of Stages | Number of Templates | Required Memory |
|---|---|---|---|
| $11 \times 17$ | 15 | 32 | 359,040 bytes |
| $21 \times 33$ | 12 | 32 | 1,064,448 bytes |
| $41 \times 65$ | 9 | 32 | 3,070,080 bytes |

Table 4.4: DSSA basis function memory requirements.

Table 4.4 gives the DSSA basis function memory requirement for each target class. Thus, the total memory for a single class is about 4.5 megabytes. For all three classes, the template memory storage is about 15 megabytes.

### DSSA Thresholds Memory Requirements

The DSSA classifier also requires memory for a threshold that is used in the decision making process. The DSSA structure is capable of representing many more patterns than are actually used in training. For example, with $N = 16$ and $P = 15$, the total number of distinct patterns that could be represented is $N^P \approx 1.0^{18}$. Clearly, only a much smaller subset of this possibility is used.

During the training of the DSSA classifier, 200 or 600 examples of SAR target signatures were used for each target. If all these training examples were sufficiently distinct, each one would produce a path through a DSSA classifier "decision tree". The number of unique paths through the tree would then equal the number of training vectors, where each path would have a "node" at each stage of the classifier. Since, typically, the average number of stages per path in these experiments is usually (as determined by experiment) about one half the maximum number of stages, the number of nodes in the tree should be upper bounded by:

$$\begin{aligned} \text{(average number of nodes)} &= \text{(number of paths)(average path length)} && (4.2)\\ &= (600)(15/2) && (4.3)\\ &= 4,500 \text{ nodes} && (4.4) \end{aligned}$$

These nodes are stored in computer memory using linked lists. The link lists have "children" which point to descendents in the linked list. Let $\mathcal{P}$ be the average number of stages per path, $c$ be the average number of children per node too estimate the number of children per node. The number of paths represented is

$$c^{\mathcal{P}} = 4,500 \tag{4.5}$$

It is clear that $c$ is the $\mathcal{P}$th root of 4,500. For an average $\mathcal{P} = 8$, the average number of children per node is $c = 2.86$

This allows an estimate the memory required for the storage of the decision tree. Conservatively estimating an average of $3 \times 2.86 \approx 9$ links per node, the node memory is given

| Feature Vector Size | Number of Stages | Number of Templates | Required Operations |
|---|---|---|---|
| $11 \times 17$ | 15 | 32 | 269,280 operations |
| $21 \times 33$ | 12 | 32 | 798,336 operations |
| $41 \times 65$ | 9 | 32 | 2,302,560 operations |

Table 4.5: DSSA basis function computation requirements.

by

$$\begin{aligned} \text{memory per node} &= 4 \text{ bytes for one threshold value} + && (4.6)\\ &\quad 36 \text{ bytes for links to 9 children} + && (4.7)\\ &\quad 4 \text{ bytes overhead} && (4.8)\\ &= 44 \text{ bytes per node} && (4.9) \end{aligned}$$

Thus, the total number of bytes used for the storage of the decision tree is

$$\begin{aligned} (\text{number of nodes in tree}) \times (\text{memory per node}) &= (4,500)(44) && (4.10)\\ &= 198,000 \text{ bytes.} && (4.11) \end{aligned}$$

### 4.3.2 Computation Requirements

The DSSA classifier must be implementable with a reasonable computational load to be practical. The number of operations required to calculate the mean squared error between an input feature and one DSSA templates is $3k$ multiplies and adds. The number of operations needed to find the most suitable DSSA basis vector at a given stage with $N$ basis functions is $3k \times N$ multiplies and adds. Thus, if $P$ stages (on average) of the DSSA classifier are used for each DSSA input, then the required number of operations is $3kNP$ multiplies and adds per DSSA decision.

One feature vector is formed for each extracted SAR snippet. Table 4.5 shows the number of operations per test snippet, as determined by the formula $3kNP$. Most snippets at the low resolution level do not result in detections, hence if overhead increases the count by about 1/2 to 400,000 operations, then "NO"-detection results cost about 0.4 million floating point operations (MFLOPS). The total, including overhead, for a snippet that is processed at all three resolution levels is about 5 MFLOPS per "YES"-detection/classification decision for each target class.

## 4.4 Reduced SAR Computation Requirements

The processing requirements of DSSA are a function of both the size of the training subregions and the number of pixels in the SAR image to be processed. The intent here is not

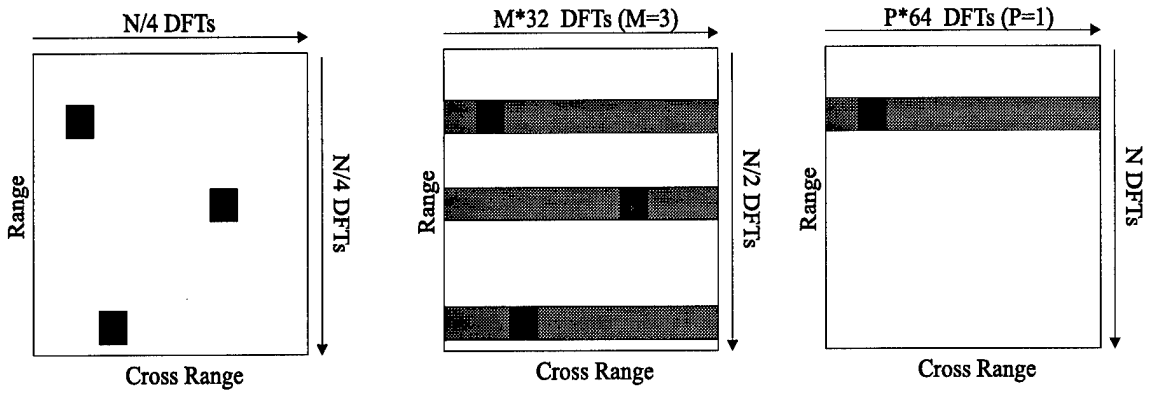


Figure 4.25: Progressive FFT-based SAR image formation.

to add to the computational requirements for forming the SAR image even though SAR images are formed at three different resolutions. The reduction in computational requirements is achieved by compressing in the cross range dimension only those with ranges associated with a potential target identified at the previous resolution. To illustrate the computational savings in this approach, assume that a DFT is used to compress the phase history data in both the range and cross range dimensions. In this example, assume a training chip size of $64 \times 64$. Assume for the moment that the phase history data consists of $K$ range samples and $K$ pulses (or cross range samples). The computational requirements to compute the DFT on a set of I&Q values is $5K\log_2(K)$ when a radix 2 FFT is used. Now to form the lowest resolution SAR image which has been degraded by a factor of four would require a DFT of size $K/4$ to be implemented $K/4$ times in range dimension and $K/4$ times in the cross range dimension resulting, in $K/2(5K/4\log_2(K/4)$ computations. After processing the SAR images at the lowest resolution, regions will be identified which contain possible targets (see Figure 4.25). These regions will be tagged and the associated range and cross range coordinates recorded. At the next level of processing, only those ranges associated with a tagged region will be compressed in cross range at the next resolution. The number of computations at the next level is

$$K/2(5K/2\log_2(K/2) + 32L(5K/2\log_2(K/2)$$

where $L$ is number of target regions identified at the lowest resolution level and 32 is the size of the training snippet at the highest resolution. The computational cost at the highest resolution is defined by

$$K/2(5K/2log2(K/2) + 64J(5K/2log2(K/2)$$

where $J$ is number of target regions identified at the medium resolution level and 64 is the size of the training snippet at the second resolution level. In this analysis it was found that $L$ and $J$ are on the order of 5 and 1 respectively.

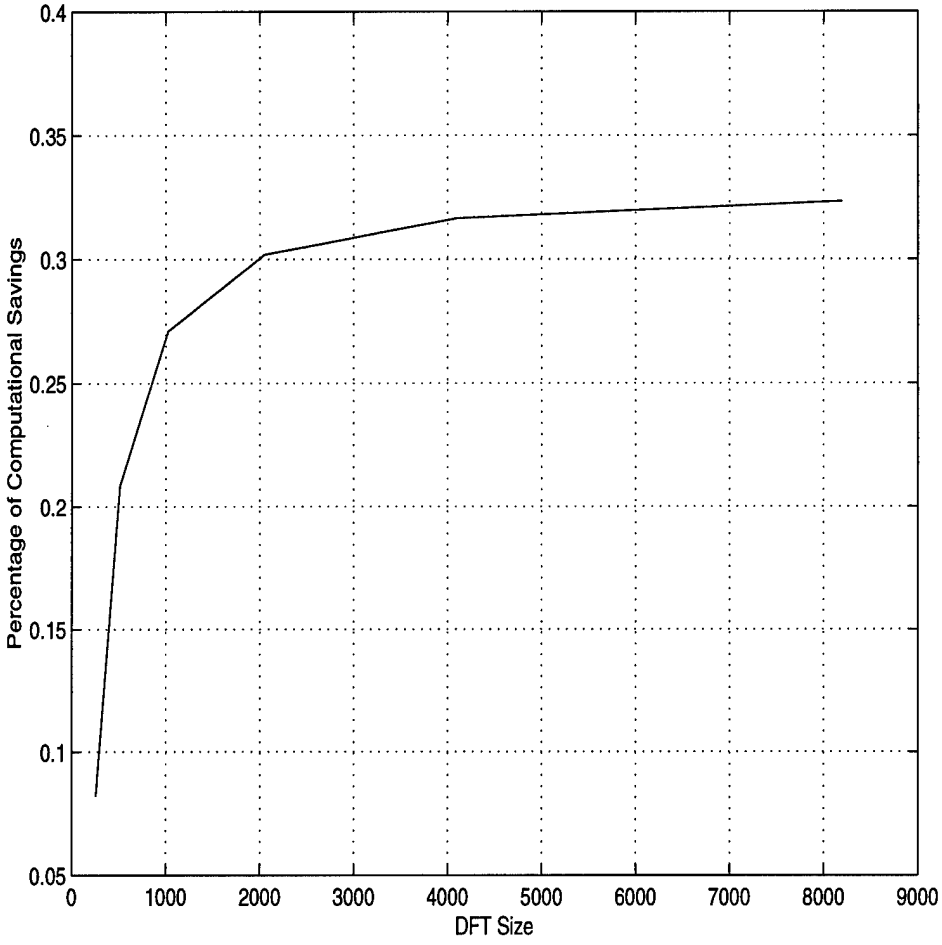


Figure 4.26: Computation savings using simple FFT-based SAR focusing.

Figure 4.26 is a plot of the computational savings in a DFT based compression approach. Thus, the SAR processing savings is about 1/3 for simple FFT image formation (the "rectangular" SAR image formation algorithm). A similar analysis of more sophisticated SAR formation algorithms such as polar formating or range-migration (omega-k) algorithms is likely to show a much more significant computational savings.

# Chapter 5

# Conclusions and Recommendations

This chapter presents GTRI's conclusions and observations. Topics for future research are also suggested.

## 5.1 Conclusions

The experimental results of this research establish that good target detection/classification can be obtained in SAR imagery with the use of direct sum exemplars formed with DSSA basis functions in a nearest neighbor classifier.

Little work was expended in this phase of the research effort to assess the compression performance of the DSSA system in the SAR application. This is scheduled as future work.

GTRI would like to emphasize that nearly all (if not indeed all) data representations used by other researchers for automatic target recognition can be viewed as feature extraction operations that aim to preserve the essential information about the signal while reducing the dimensionality of the data [66]. The DSSA approach presented in this report does not conform to this paradigm—DSSA preserves the essential information about the signal in a compute and memory efficient manner while maintaining the full dimensionality of the data. This research establishes the feasibility of this approach with SAR data for target detection and classification, and thus opens the door for future research that exploits the generality and flexibility of DSSA.

## 5.2 Observations

GTRI believes that one reason for a high level of correct classification, even with the use of limited training data in a high dimensional decision space, is the progressive "bootstrapping" performed by the direct sum exemplars as additional DSSA basis functions are added in the

search process. Bootstrapping is the process of expanding a limited training set by locally combining the original training samples, where various numbers of near-neighbors are used [67]. Bootstrapping acts as a smoother of the empirical distribution, and softens the negative effects of outliers. Each path through the DSSA basis functions specifies a different local collection of training samples, and the basis functions are a causal-anti-causal centroid of this local collection [3]. Hence, the process of forming direct sum exemplars, in effect, bootstraps the training set.

## 5.3 Suggested Future Research and Development

The following topics are suggested for additional research and development.

### Testing of the DSSA-Detector/Classifier on EO/IR Data

GTRI would like to extend this work to testing for targets in EO/IR data. However, GTRI currently doesn't have access to data to support this work.

### Extension of the Database-Classifier Paradigm

The DSSA classifier is in essence a database classifier [66]. The direct sum exemplars provide a rapid-search interface to determine a similar entry in the of training data to a given classifier input. Since the nearest direct sum exemplar can be directly associated with a training set entry in a practical manner (a relational query), if appurtenant data exists, such as aspect angle, grazing angle, clutter type, propagation conditions, etc., this database classifier can provide this archived data for comparison to the current data, which also provide an estimate of the state of the object represented by the classifier input data. This classification system should be extended to provide this capability.

### Use of Improved Features

Tests to date have used blocks of SAR pixels as input. This approach does not rely on the manual definition of a more complex feature set, and it captures all available information in the data space. However, the DSSA architecture also permits the use of many other kind of features. The DSSA algorithm is not restricted to raw sample inputs. Tests should be performed to determine whether a hybrid approach can be used that includes both raw data input as well as other features that are known by MSTAR to be helpful in discriminating targets from clutter.

### Multiple Nearest Neighbors for DSSA Classification

In the literature that deals with nearest neighbor classifiers, generalizations of single nearest neighbor classifiers to multiple nearest neighbor classifiers are common. A multiple nearest neighbor classifier uses, in a sense, local class-conditional template density as a basis for discrimination. A multiple nearest neighbor generalization of the DSSA classifier can be easily accomplished with the use of multiple path searching of the sequence of DSSA basis functions. GTRI has conducted extensive prior research into multiple path search techniques in the area of data compression [68, 69], this research is yet to be extended into the data classification area.

# Appendix A

# Various SAR Target Configurations

Photographic images of the some of the configuration of target vehicles are shown in Figures A.2–A.3.

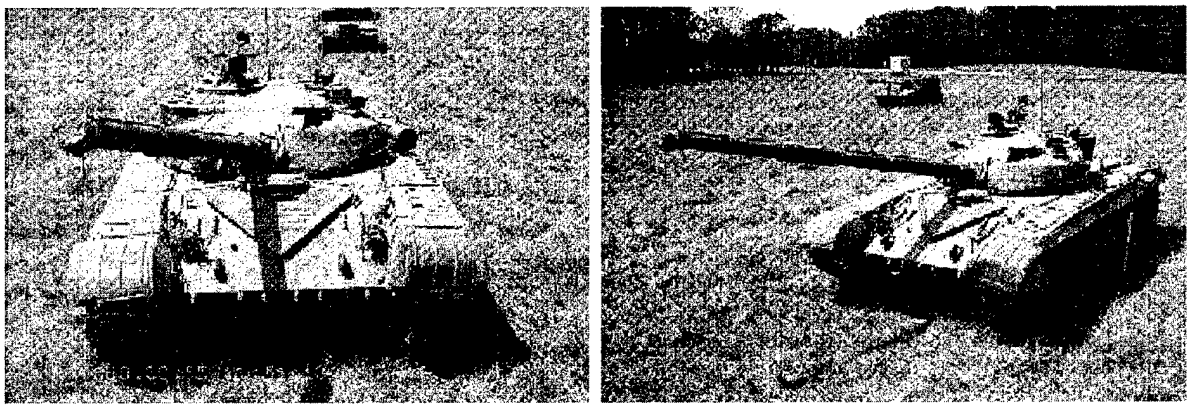

Figure A.1: T72 tank in S7 configuration.

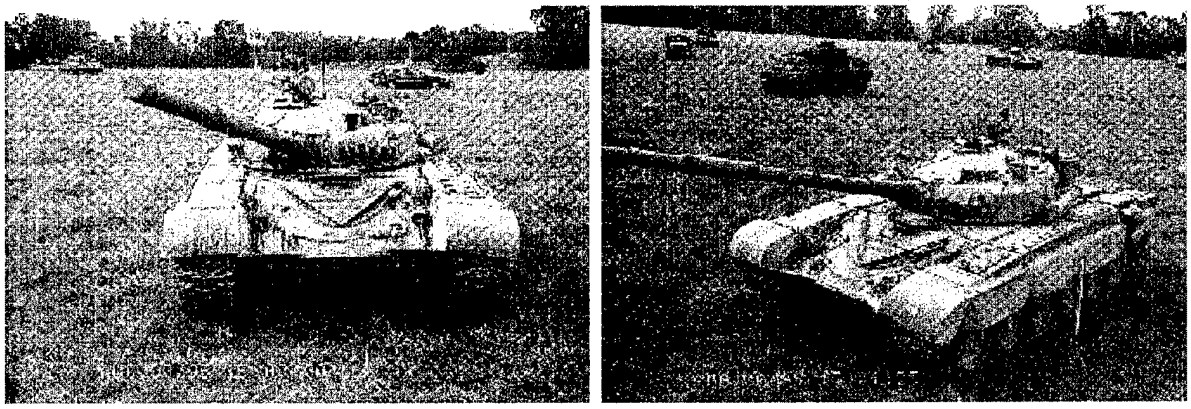

Figure A.2: T72 tank in 132 configuration.

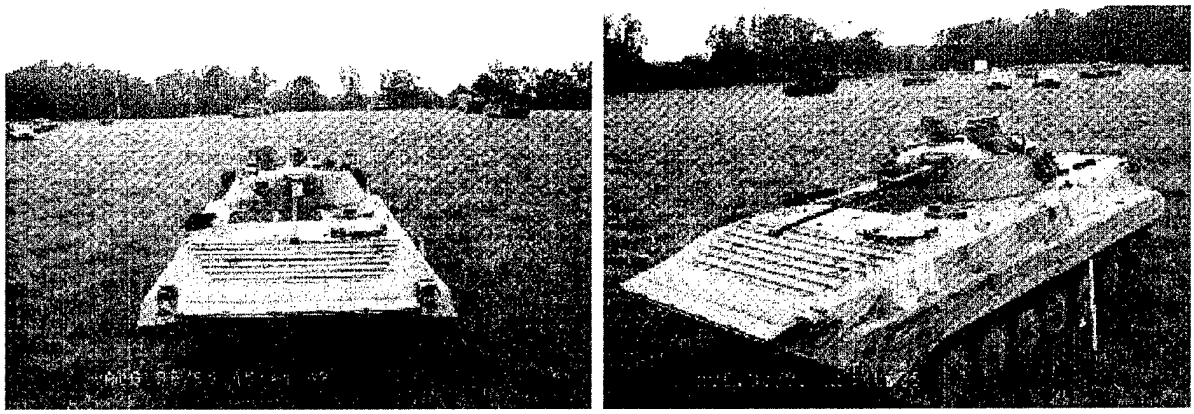

Figure A.3: BMP2 infantry fighting vehicle in 9566 configuration.

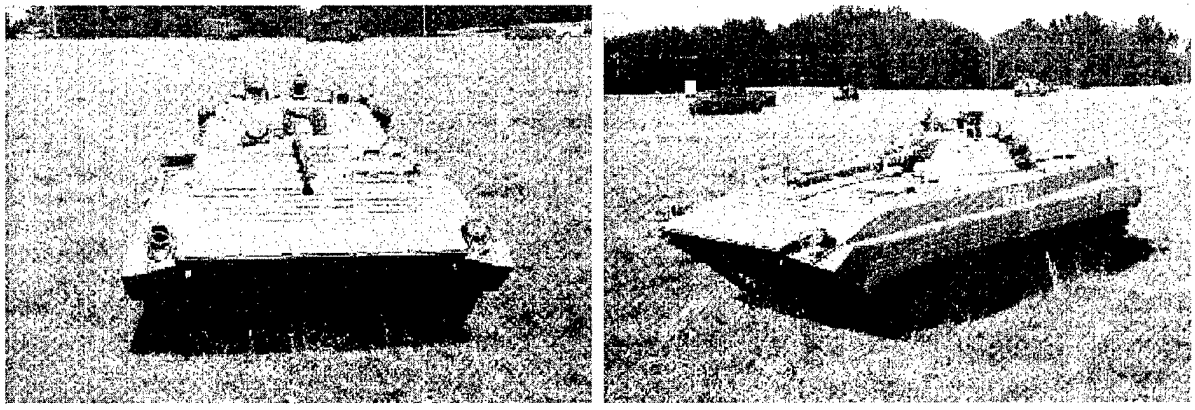

Figure A.4: BMP2 infantry fighting vehicle in 9563 configuration.

# Appendix B

# Example SAR Clutter Images

Example SAR images of clutter scenes are shown Figures B.1–B.4.

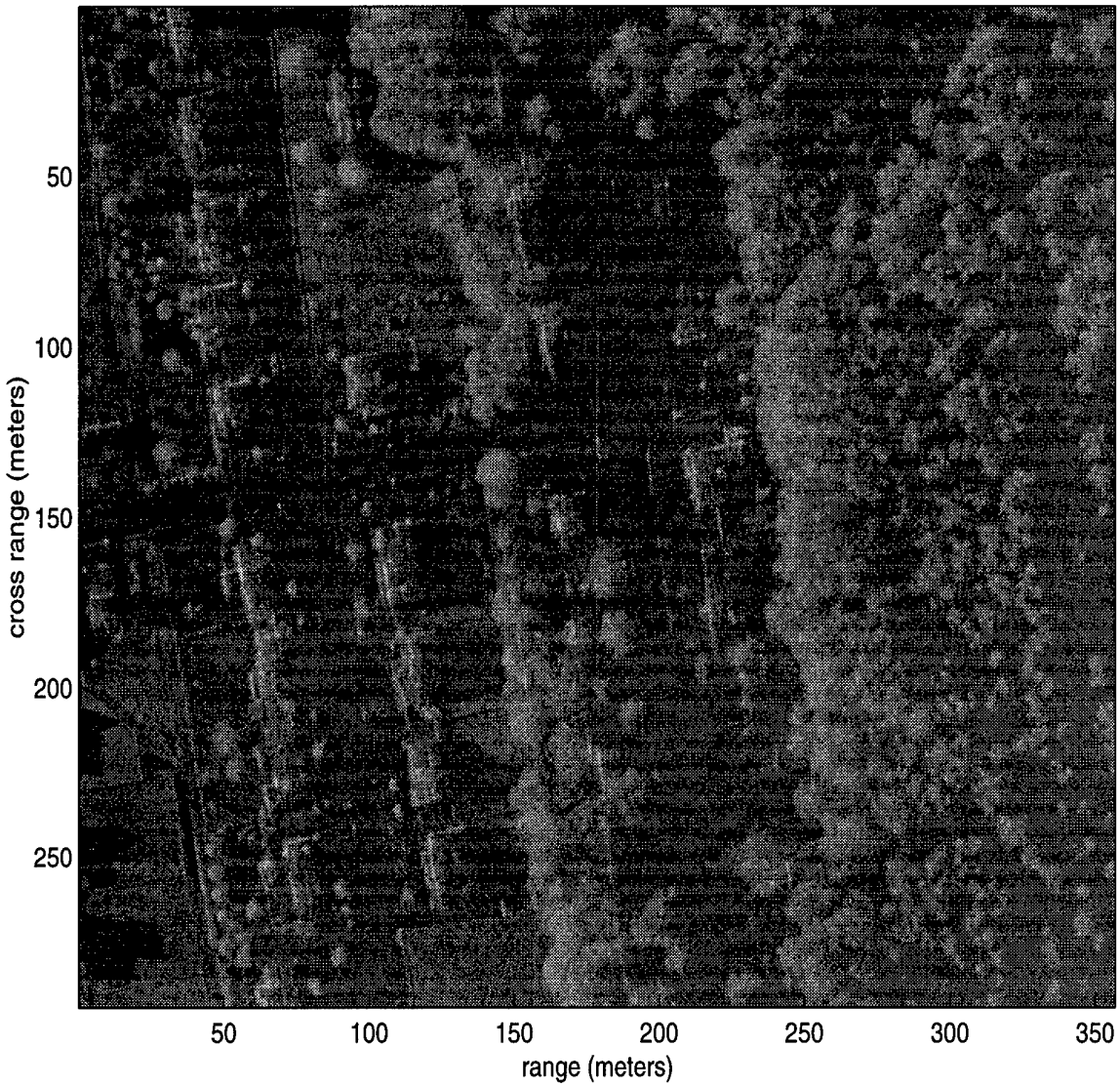


Figure B.1: Example high resolution SAR clutter image.

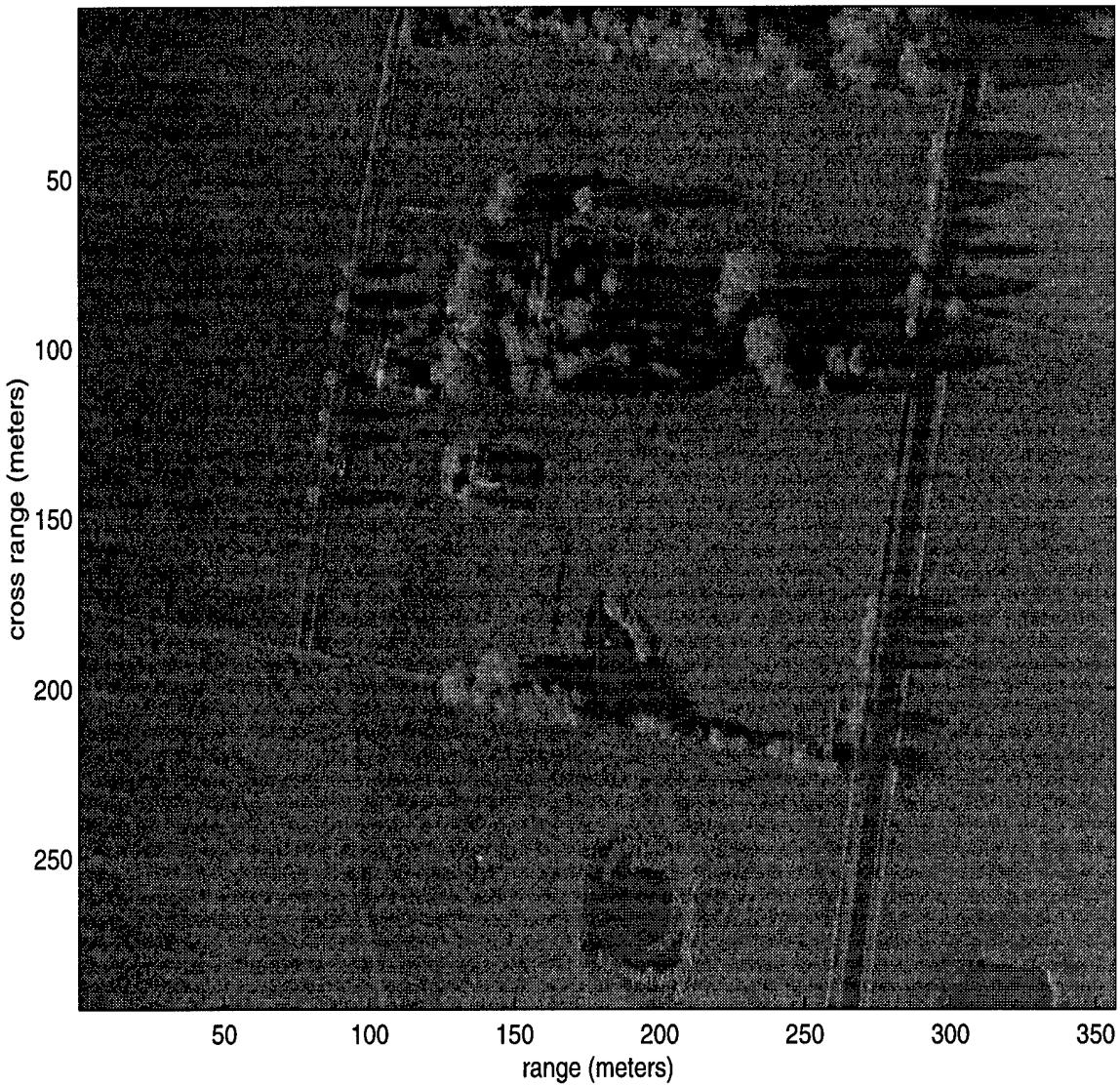


Figure B.2: Example high resolution SAR clutter image.

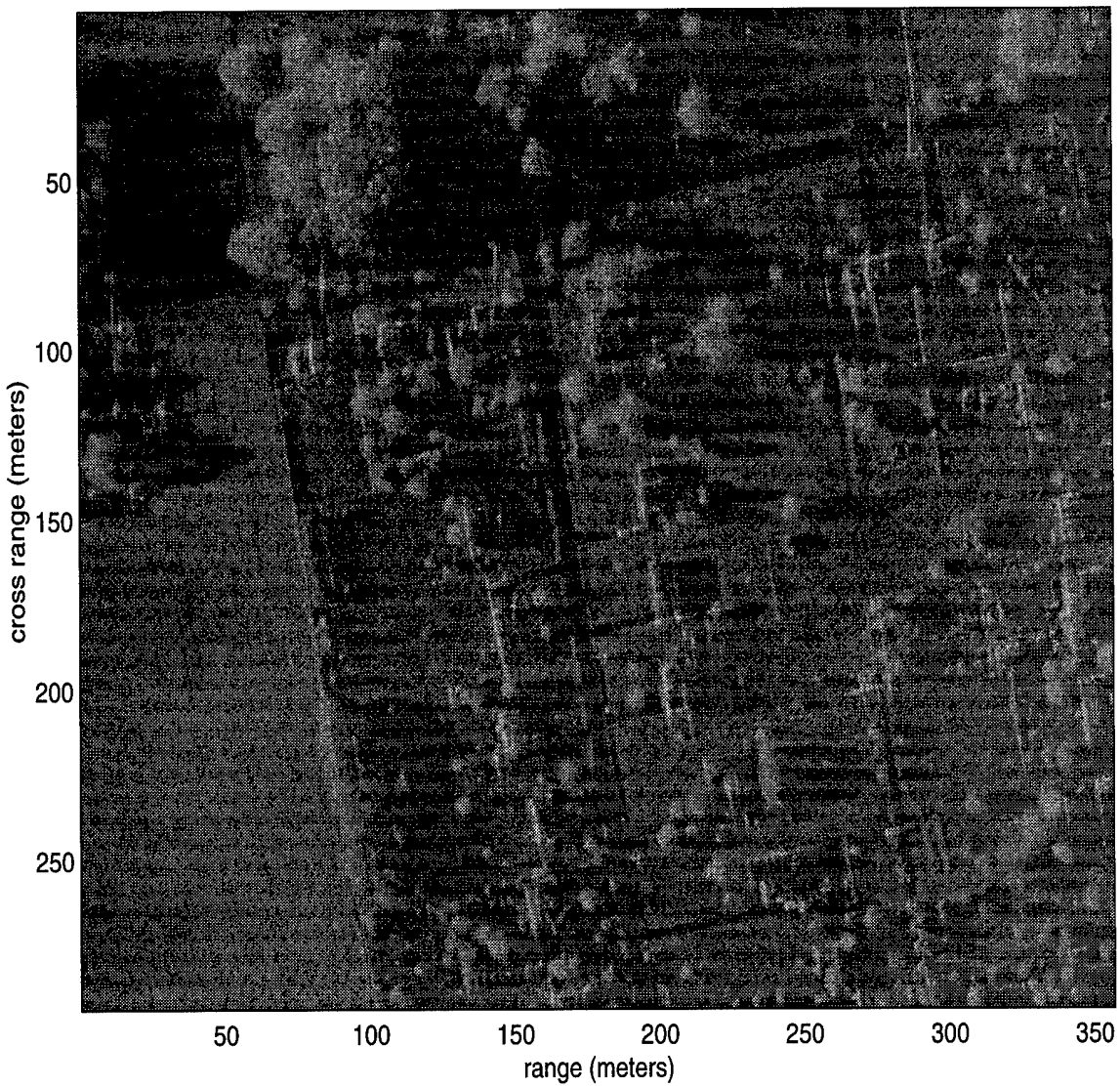


Figure B.3: Example high resolution SAR clutter image.

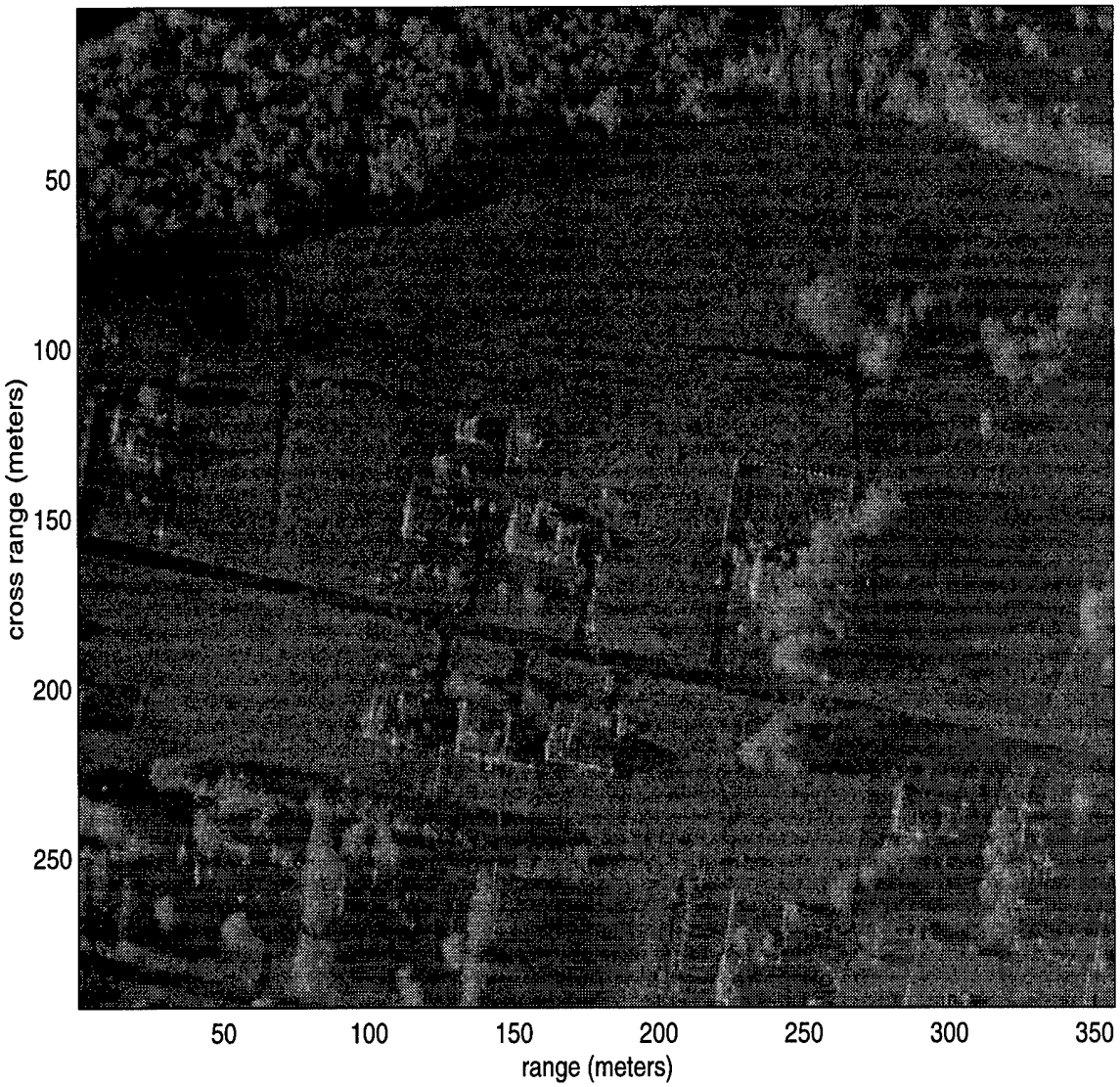


Figure B.4: Example high resolution SAR clutter image.

# Appendix C

# List of Abbreviations

This list provides expansions for abbreviations and acronyms used in this report.

| Abbreviation/Acronym | Explanation |
|---|---|
| ATR | Automatic Target Recognition |
| DARPA | Defense Advanced Research Project Agency |
| DBF | DSSA Basis Function |
| DoF | Degrees of Freedom |
| DSSA | Direct Sum Successive Approximation |
| DoD | Department of Defense |
| EO | Electro-Optical |
| FLS | Forward Looking Sonar |
| GTRI | Georgia Tech Research Institute |
| HAE | High Altitude Endurance |
| IR | Infrared |
| MS | Multispectral |
| MSVQ | Multiple Stage Vector Quantization |
| MSTARS | Moving and Stationary Target Acquisition |
| MTI | Moving Target Indication |
| NN | Nearest Neighbor |
| NNet | Neural Net |
| NN-RVQ | Nearest Neighbor Residual Vector Quantization |
| ONR | Office of Naval Research |
| RBF | Radial Basis Function |
| ROI | Region-of-Interest |
| SAR | Synthetic Aperture Radar |
| SIGINT | Signal Intelligence |
| SLS | Side Looking Sonar |
| RVQ | Residual Vector Quantization |
| UAV | Unmanned Aerial Vehicles |
| VQ | Vector Quantization |

# Bibliography


[1] C. F. Barnes. Feasibility studies of nearest neighbor residual vector quantizer classifiers for a collection of signal and sensor waveforms: Computer aided diagnosis of mammography. GTRI Interim Technical Report Project No. A-5138, Georgia Tech Research Institute, Atlanta, GA, January 1, 1996.

[2] B. H. Juang and A. H. Gray. Multiple stage vector quantization for speech coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 597–600, April 1982.

[3] C. F. Barnes. *Residual Quantizers.* PhD thesis, Brigham Young University, Provo, UT, December 1989.

[4] W.-Y. Chan. *Product Code Vector Quantization Methods with Application to High Fidelity Audio Coding.* PhD thesis, University of California, Santa Barbara, December 1991.

[5] C. F. Barnes and R. L. Frost. Necessary conditions for the optimality of residual vector quantizers. In *Abstracts of the IEEE International Symposium on Information Theory*, page 34, San Diego, CA, January 14–19, 1990.

[6] C. F. Barnes and R. L. Frost. *Image Mathematics and Image Processing*, volume 84 of *Advances in Electronics and Electron Physics*, chapter Residual Vector Quantizers with Jointly Optimized Code Books, pages 1–59. Academic Press, 1992. Edited by P. W. Hawkes.

[7] C. F. Barnes and R. L. Frost. Vector quantizers with direct sum codebooks. *IEEE Transactions on Information Theory*, 39(2):565–580, March 1993.

[8] C. F. Barnes, S. Rizvi, and N. Nasrabadi. Residual vector quantization: A review. *IEEE Transactions on Image Processing*, February 1996.

[9] U.S. Patent 5,250,949: *System and Method for Data Compression Using Multiple Codewords and Transmitted Indices*, Filed 14 August 1989, Issued 5 October 1993.

[10] E. E. Hilbert. Cluster compression algorithm: A joint clustering/data compression concept. Publication 77–43, Jet Propulsion Laboratory, Pasadena, CA, December 1977.

[11] K. L. Oehler, P. C. Cosman, R. M. Gray, and J. May. Classification using vector quantization. In *Proceedings of the Twenty-Fifth Annual Asilomar Conference on Signals, Systems, and Computers*, pages 439–445, Pacific Grove, CA, November 1991.

[12] K. L. Oehler and R. M. Gray. Combining image classification and image compression using vector quantization. In J. A. Storer and M. Cohn, editors, *Proceedings of the Data Compression Conference*, pages 2–11, Snowbird, UT, March 30–April 2 1991.

[13] R. M. Gray, K. L. Oehler, K. O. Perlmutter, and R. A. Olshen. Combining tree-structured vector quantization with classification and regression trees. In *Proceedings of the Twenty-Seventh Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 1–3, 1993.

[14] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Berkeley Symposium on Math. Stat. and Prob.*, volume 1, pages 281–297, 1967.

[15] S. P. Lloyd. Least squares quantization in PCM. *Bell Laboratories Technical Note*, 1957. Published in the March 1982 Special Issue on Quantization: *IEEE Transactions on Information Theory, Part 1*, 28:129–137.

[16] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(2):7–12, March 1960.

[17] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, January 1980.

[18] R. M. Gray. Vector quantization. *IEEE Acoustics, Speech and Signal Processing Magazine*, 1:4–29, April 1984.

[19] T. Kohonen. An introduction to neural computing. *Neural Networks*, 1:3–16, 1988.

[20] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, Germany, third edition, 1989.

[21] A. LaVigna. *Nonparametric Classification Using Learning Vector Quantization.* PhD thesis, University of Maryland, 1989.

[22] C. F. Barnes. An initial evaluation of a new approach to high resolution sonar imagery. GTRI Final Technical Report Project No. A-9523, Georgia Tech Research Institute, Atlanta, GA, November 1994.

[23] C. F. Barnes. Continued development of unique mathematical and statistical signal processing algorithms for detecting and classifying mines in acoustic backscatter. GTRI Interim Technical Report Project No. A-5258, Georgia Tech Research Institute, Atlanta, GA, January 1998.

[24] G. F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, 1968.

[25] K. Fukunaga and D. M. Hummels. Bias of nearest neighbor error estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):103–112, January 1987.

[26] G. Lugosi and K. Zeger. Concept learning using complexity regularization. *IEEE Transactions on Information Theory*, 42(1):48–54, January 1996.

[27] S. Raudys. A bootstrap technique for nearest neighbor classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):73–79, January 1997.

[28] S. Raudys. On dimensionality, sample size, and classification error of nonparametric linear classification algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):667–671, June 1997.

[29] E. Fix and J. L. Hodges, Jr. Discriminatory analysis, nonparametric discrimination, consistency properties. Report 4 on Project 21–49–004, USAF School of Aviation Medicine, Randolph Field, TX, February 1951.

[30] E. W. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21(3):768, abstract, 1965.

[31] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, January 1967.

[32] G. F. McLean. Texture classification using transform vector quantization. In *Proc. SPIE–The Int. Soc. Optical Engineering: Visual Communications and Image Processing*, volume 1360, pages 1332–1343, Lausanne, Switzerland, 1990.

[33] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray. Using vector quantization for image processing. *Proceedings of the IEEE*, 81(9):1326–1341, September 1993.

[34] L. M. Owsley and L. E. Atlas. Ordered vector quantization for neural network pattern recognition. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, 1993.

[35] A. Duchon and S. Katagiri. A minimum-distortion segmentation/LVQ hybrid algorithm for speech recognition. *J. Acoust. Soc. Japan (E)*, 14(1):37–42, 1993.

[36] J. J. Merelo, M. A. Andrade, C. Urena, A. Preito, and F. Moran. Application of vector quantization for protein classification and secondary structure computation. In *Artificial Neural Networks. International Workshop IWANN'91 Proc.*, pages 415–421, New York, NY, 1991. Springer-Verlag.

[37] J. Ghos, S. Chakravarthy, Y. Shin, C.-C. Chu, L. Deuser, S. Beck, R. Still, and J. Whiteley. Adaptive kernel classifiers for short-duration oceanic signals. In *Proc. IEEE Conf. on Neural Networks for Ocean Engineering*, pages 41–48, 1991.

[38] R. O. Harger. Object detection in clutter with learning maps. In *Synthetic Aperture Radar*, pages Proc. SPIE 1630, 176–186, 1992.

[39] E. K. Neumann, D. A. Wheeler, A. S. Bernstein, and J. W. Burnside. Artificial neural network classification of Drosophila courtship song mutants. *Bilogical Cybernetics*, 66(6):485–496, 1992.

[40] K. Reiser. Vector quantization for recognition of hand written numerals. In *Proceedings of the Twenty-Seventh Annual Asilomar Conference on Signals, Systems, and Computers*, volume 2, pages 1646–1650, Pacific Grove, CA, November 1–3, 1993.

[41] L. Bruzzone, F. Roli, and S. B. Serpico. An extension of the jeffreys-matsushita distance to multiclass cases for feature selection. *IEEE Transactions on Geoscience and Remote Sensing*, 33(6):1318–1321, November 1995.

[42] I. Csiszar. Information-type measures of difference of probability distributions and indirect observations. *Studia. Scientarium Mathematicerium Hungarica*, 2:229–318, 1967.

[43] H. V. Poor and J. B. Thomas. Applications of Ali-Silvey distance measures in the design of generalized quantizers for binary decision systems. *IEEE Transactions on Communications*, 25(9):893–900, September 1977.

[44] H. V. Poor. Robust decision design using a distance criterion. *IEEE Transactions on Information Theory*, 26(5):575–587, September 1980.

[45] H. V. Poor. Robust quantization of $\epsilon$-contaminated data. *IEEE Transactions on Communications*, 33:218–222, 1985.

[46] H. V. Poor. Fine quantization in signal detection and estimation. *IEEE Transactions on Information Theory*, 34(5):960–972, September 1988.

[47] J. N. Tsitsiklis. Extremal properties of likelihood-ratio quantizers. *IEEE Transactions on Communications*, 41(4):550–558, April 1993.

[48] G. H. Martinez, C. Rivera, and A. Buzo. Discrete utterance recognition based upon source coding techniques. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 539–542, 1982.

[49] J. L. Sungook and C.-C. J. Kuo. Focus of attention (FOA) identification from compressed video for automatic target recognition (ATR). In *Proceedings of the IEEE International Conference on Image Processing*, pages 508–511, Washington, DC, October 1995.

[50] Y. Ephraim and R. M. Gray. A unified approach for encoding clean and noisy sources by means of waveform and autoregressive model vector quantization. *IEEE Transactions on Information Theory*, 34(4):826–834, July 1988.

[51] E. Ayanoğlu. On optimal quantization of noisy sources. *IEEE Transactions on Information Theory*, 36(6):1450–1452, November 1990.

[52] W.-M. Lam and A. R. Reibman. Design of quantizers for decentralized estimation systems. *IEEE Transactions on Communications*, 41(11):1602–1605, November 1993.

[53] S. A. Kassam. *Optimal Data Quantization in Signal Detection*, chapter 4, pages 72–110. Communications and Networks: A Survey of Recent Advances. Springer-Verlag, 1986. Edited by I. F. Blake and H. V. Poor.

[54] A. Wald. *Sequential Analysis.* Dover, New York, 1947.

[55] N. S. Subotic and B. J. Thelen. Sequential processing of SAR phase history data for rapid detection. In *Proceedings of the IEEE International Conference on Image Processing*, pages 144–146, Washington, DC, October 1995.

[56] A. Dembo and Y. Peres. A topological criterion for hypothesis testing. *Ann. Statist.*, 22:106–117, 1994.

[57] T. J. Flynn and R. M. Gray. Encoding of correlated observations. *IEEE Transactions on Information Theory*, 33(6):773–787, November 1987.

[58] M. Longo, T. Lookabaugh, and R. M. Gray. Quantization for decentralized hypothesis testing under communication constraints. *IEEE Transactions on Information Theory*, 36(2):241–255, March 1990.

[59] R. S. Blum. Quantization in multisensor random signal detection. *IEEE Transactions on Information Theory*, 41(1):204–215, January 1995.

[60] L. L. Burton and H. Lai. Active sonar imaging and classification system. In *SPIE*, volume 3079, Orlando, FL, 20–25 April 1997.

[61] T. Kohonen, G. Barna, and R. Chrisley. Statistical pattern recognition with neural networks: Benchmarking studies. In *IEEE International Conference on Neural Networks*, volume I, pages 61–68, July 1988.

[62] C. J. Stone. Consistent nonparametric regression. *Annals of Statistics*, 5:595–645, 1977.

[63] L. A. Chan, N. M. Nasrabadi, and V. Mirelli. Multistage target recognition using modular vector quantizers and multilayer perceptrons. In *Proceedings 1996 Computer Society Conference on Computer Vision and Pattern Recognition*, pages 114–119, San Francisco, CA, 18–20 June 1996.

[64] D. Burton. Acoustic transient classification of passive sonar signals by using vector quantization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1493–1496, Toronto, Ontario, Canada, 1991.

[65] H. Qu and J. Gotman. On the finite sample performance of the nearest neighbor classifier. *IEEE Transactions on Biomedical Engineering*, 44(2):115–122, February 1997.

[66] L. M. Garth and H. V. Poor. Detection of non-gaussian signals: A paradigm for modern statistical signal processing. *Proceedings of the IEEE*, 82(7):1061–1095, July 1994.

[67] B. Efron. Bootstrp methods: Another look at the jackknife. *Annual Statistics*, 7:1–26, 1979.

[68] C. F. Barnes. A new multiple path search technique for residual vector quantizers. In *Proceedings of the Data Compression Conference*, pages 42–51, Snowbird, UT, March 29–31, 1994.

[69] F. Kossentini, M. J. T. Smith, and C. F. Barnes. Large block RVQ with multipath searching. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 5, pages 2276–2279, San Diego, CA, May 10–13, 1992.